GPU BASED HOUGH TRACK FINDER

15.06.2021 I PANDA COLLABORATION MEETING | ANNA ALICKE

WHY USING GPUS?

Intention

- tracking: most computing intensive part of event reconstruction
- high event rate in panda: 20 mio events/s

→ tracking algorithm: must be really fast to reduce the amount of data

→ test if GPUs can deliver the speed up we need

WHY USING GPUS?

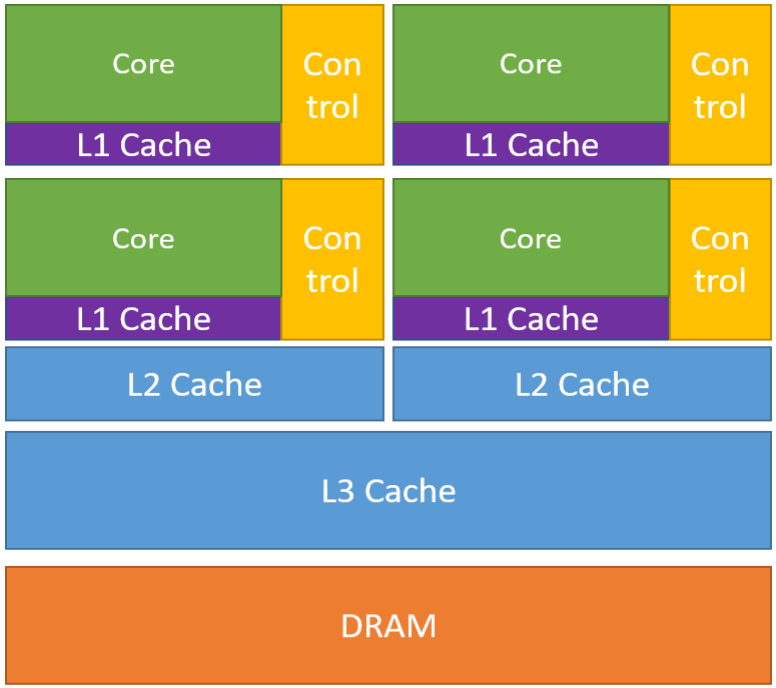
Architectural differences: CPU vs GPU

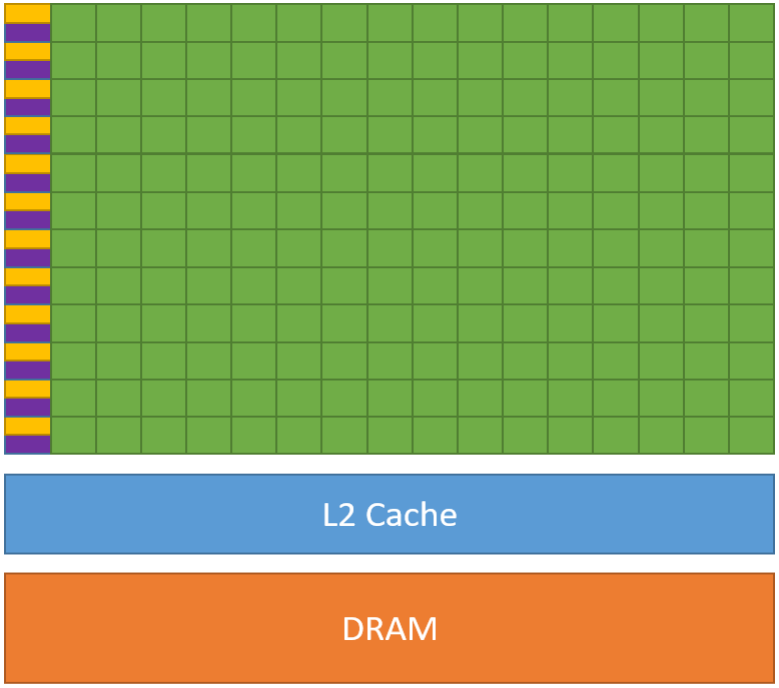
CPU

- Large cache and controlling
  → fast handling of complex sequential calculations

GPU

- Small cache and controlling
  → many small calculations in parallel with less memory accesses





Taken from: NVIDIA Corporation (2021), 'NVIDIA CUDA C++ Programming Guide' , Version 11.2 .1.

WHY USING GPUS?

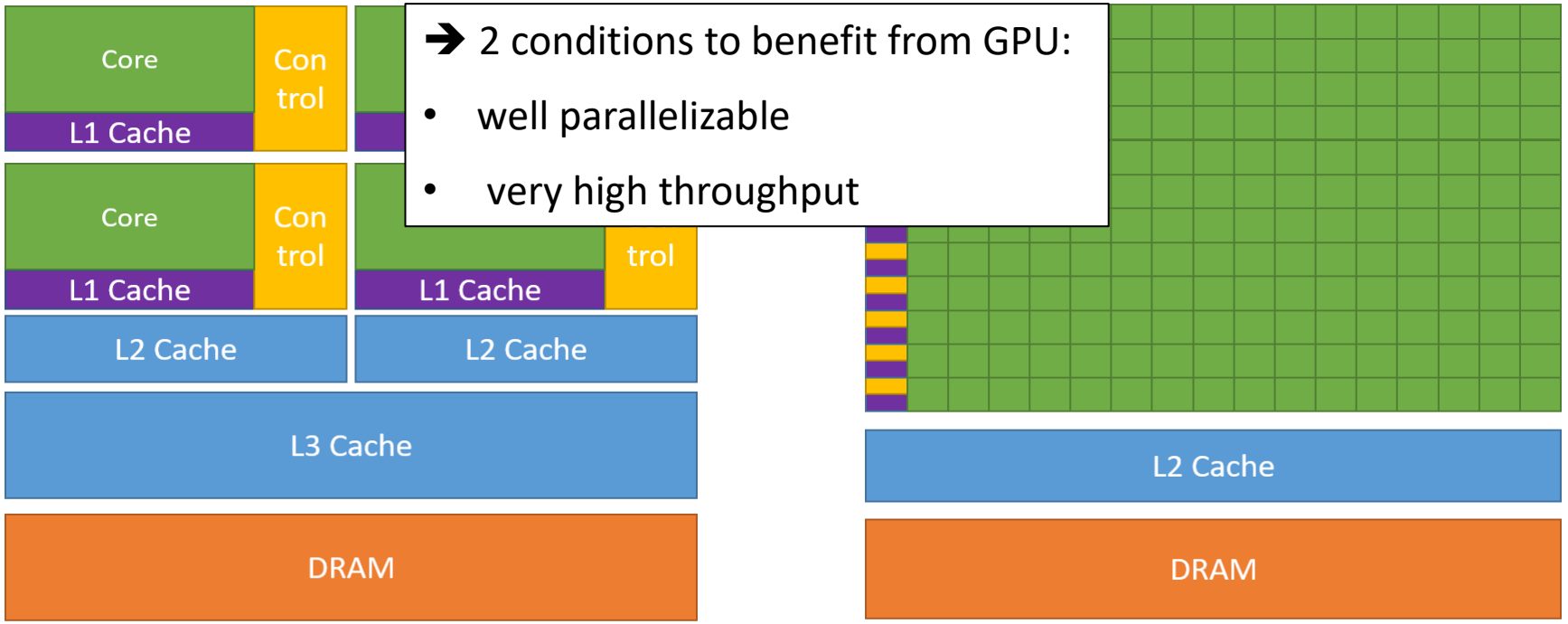
Architectural differences: CPU vs GPU

CPU

- Large cache and controlling
  → fast handling of complex sequential calculations

GPU

- Small cache and controlling
  → many small calculations in parallel with less memory accesses



➔ 2 conditions to benefit from GPU:

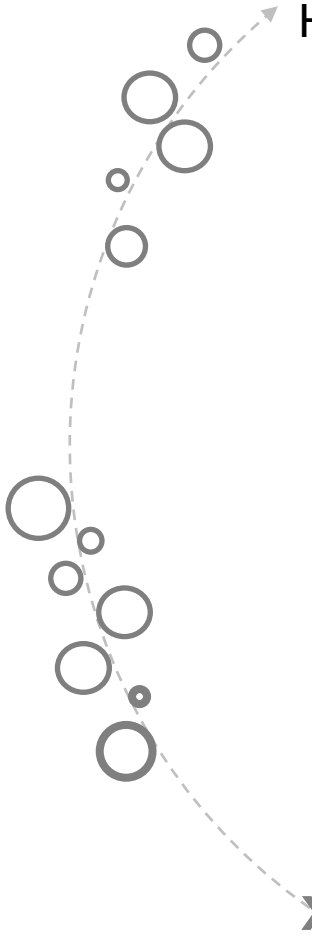
- well parallelizable
- very high throughput

Taken from: NVIDIA Corporation (2021), 'NVIDIA CUDA C++ Programming Guide' , Version 11.2 .1.

WHY USING GPUS?

Does the HoughTrackFinder fulfill the conditions?

Hough transformation + Apollonius problem

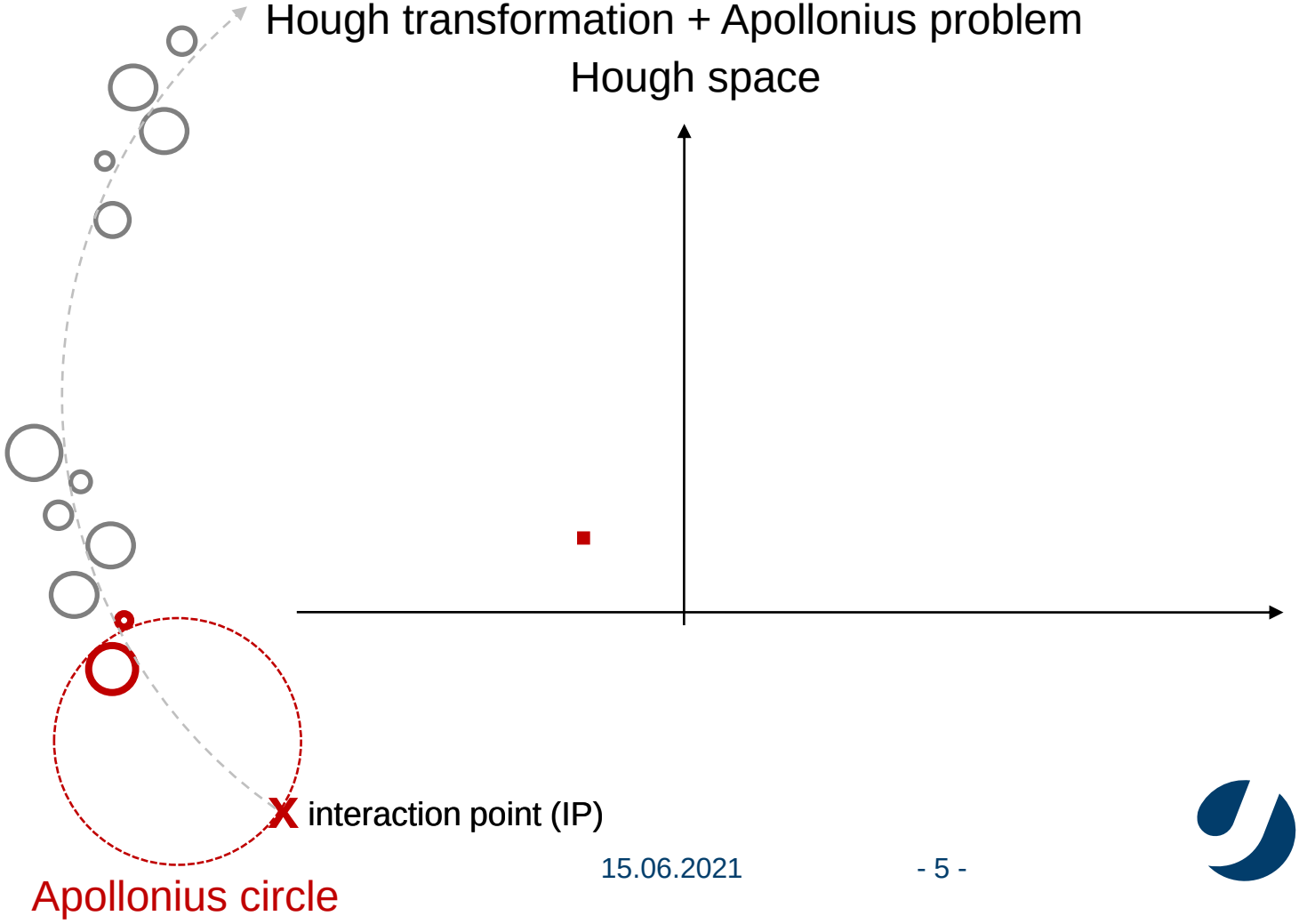


X interaction point (IP)

WHY USING GPUS?

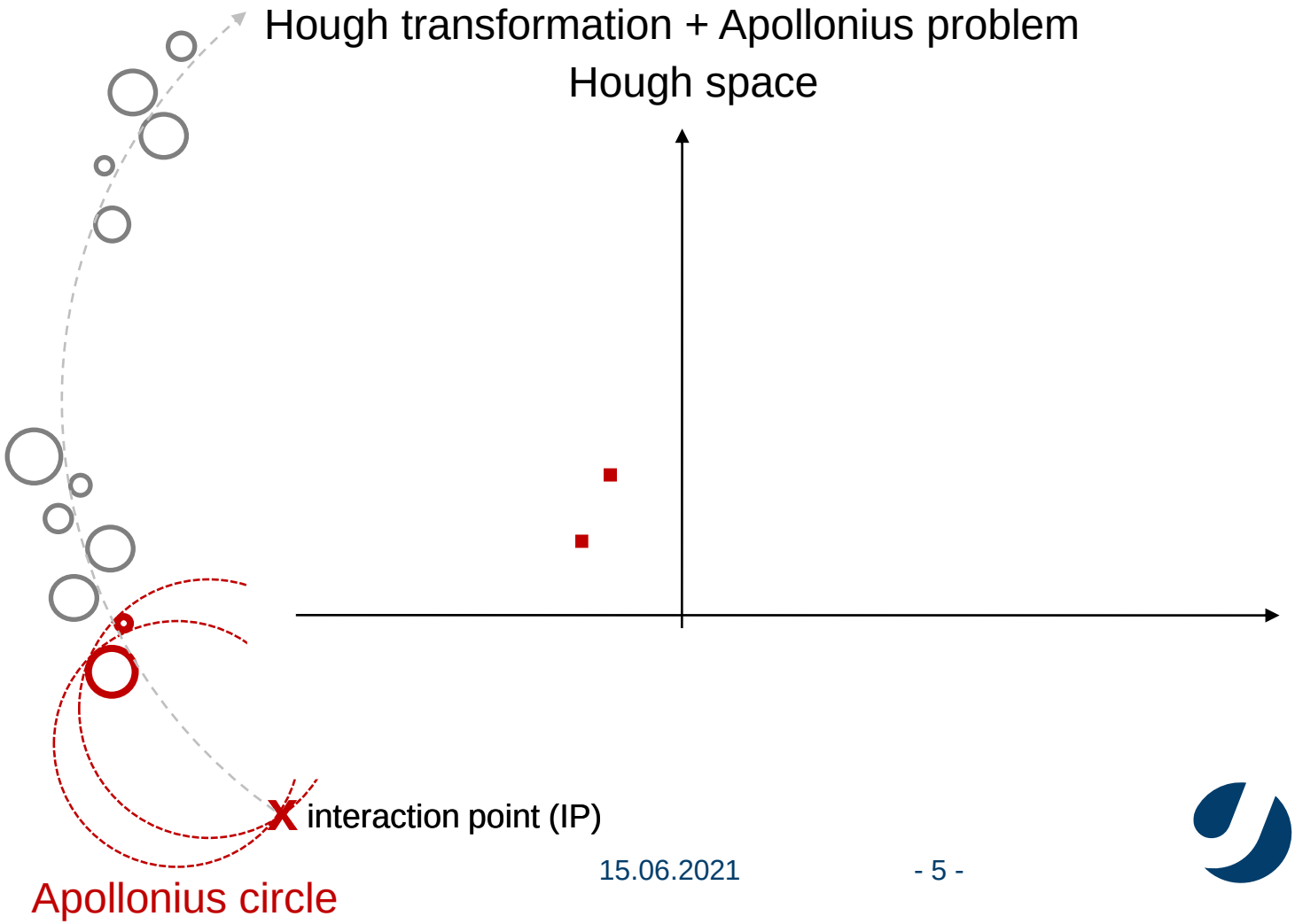
Does the HoughTrackFinder fulfill the conditions?

Hough transformation + Apollonius problem



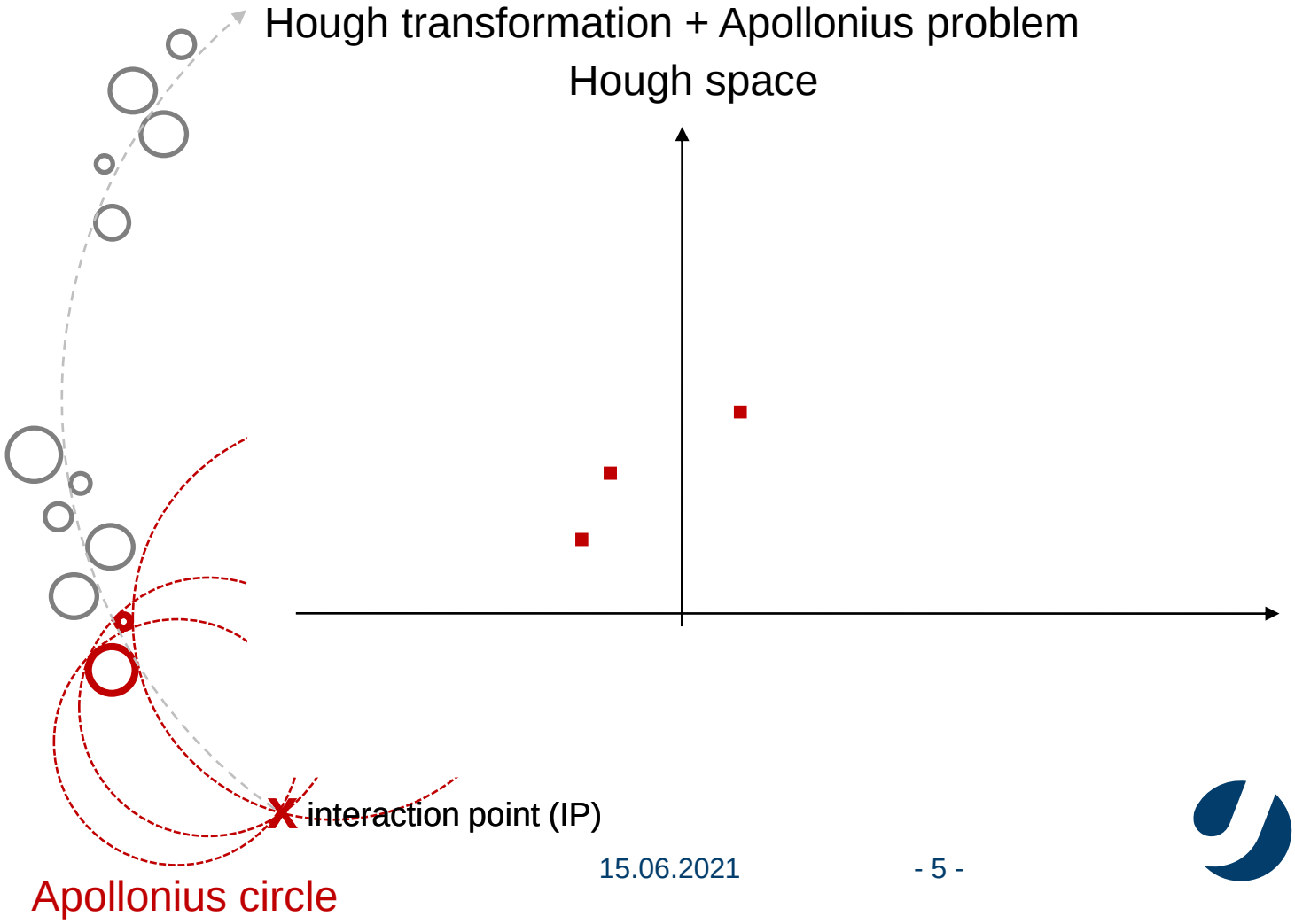
WHY USING GPUS?

Does the HoughTrackFinder fulfill the conditions?



WHY USING GPUS?

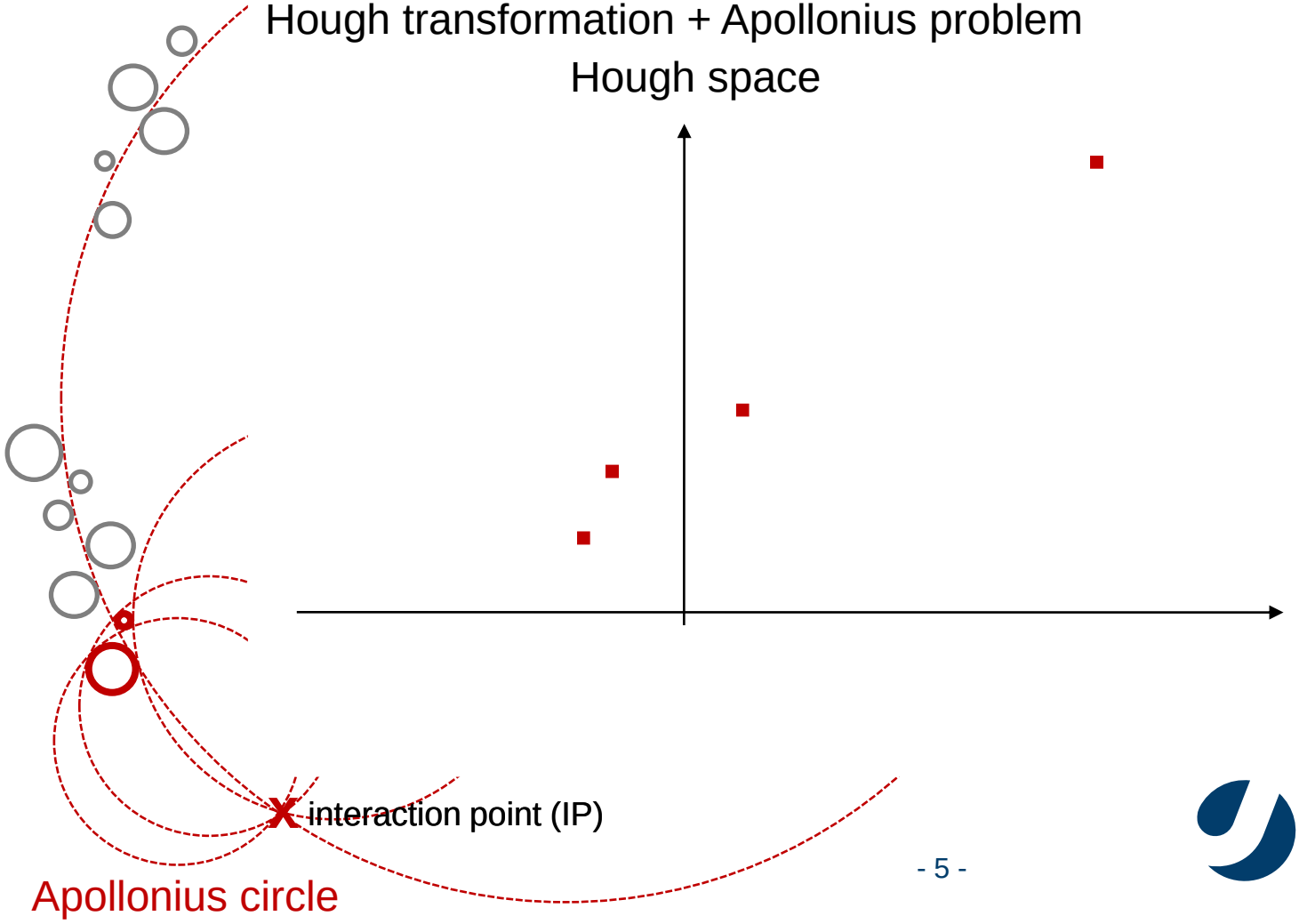
Does the HoughTrackFinder fulfill the conditions?



WHY USING GPUS?

Does the HoughTrackFinder fulfill the conditions?

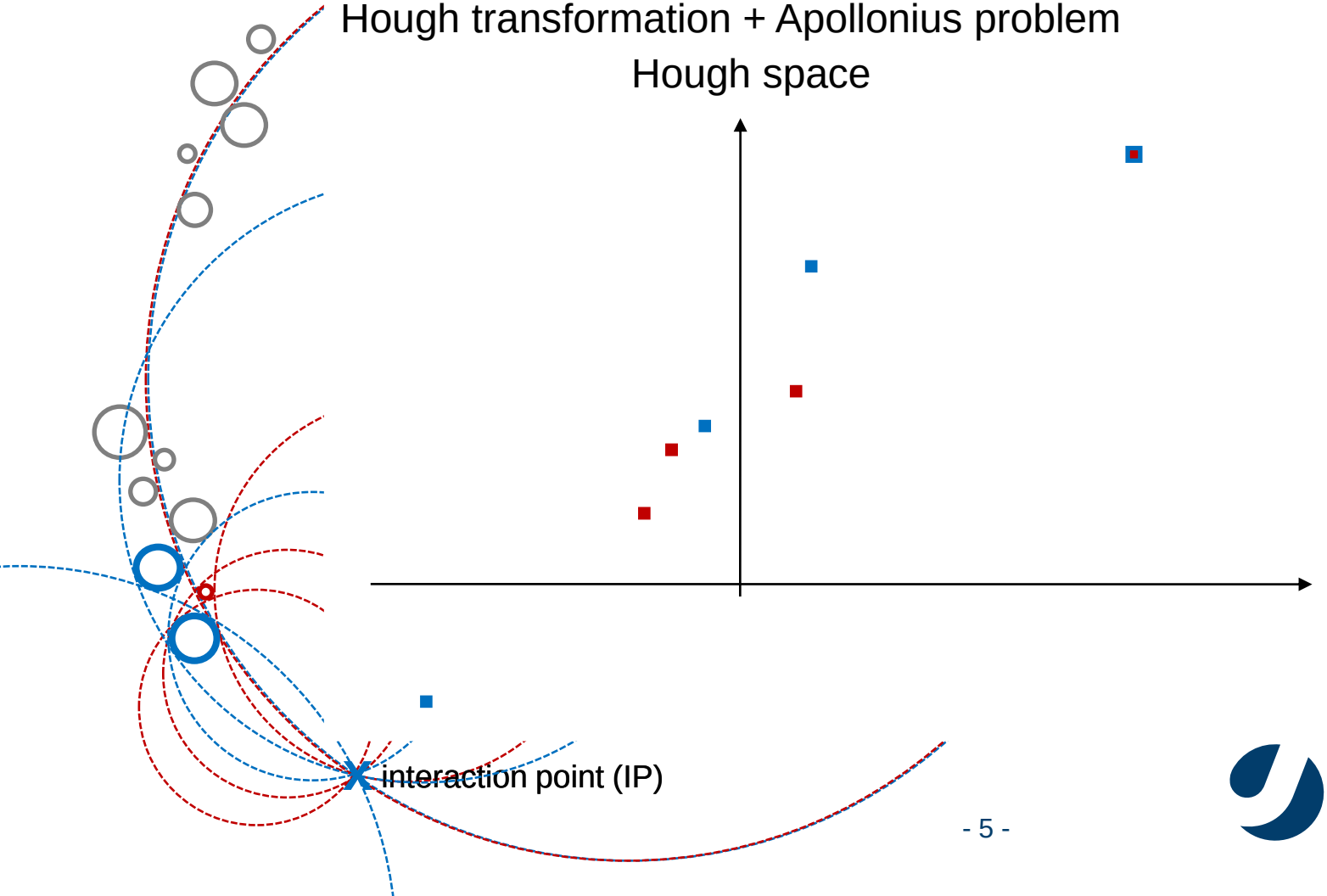
Hough transformation + Apollonius problem



WHY USING GPUS?

Does the HoughTrackFinder fulfill the conditions?

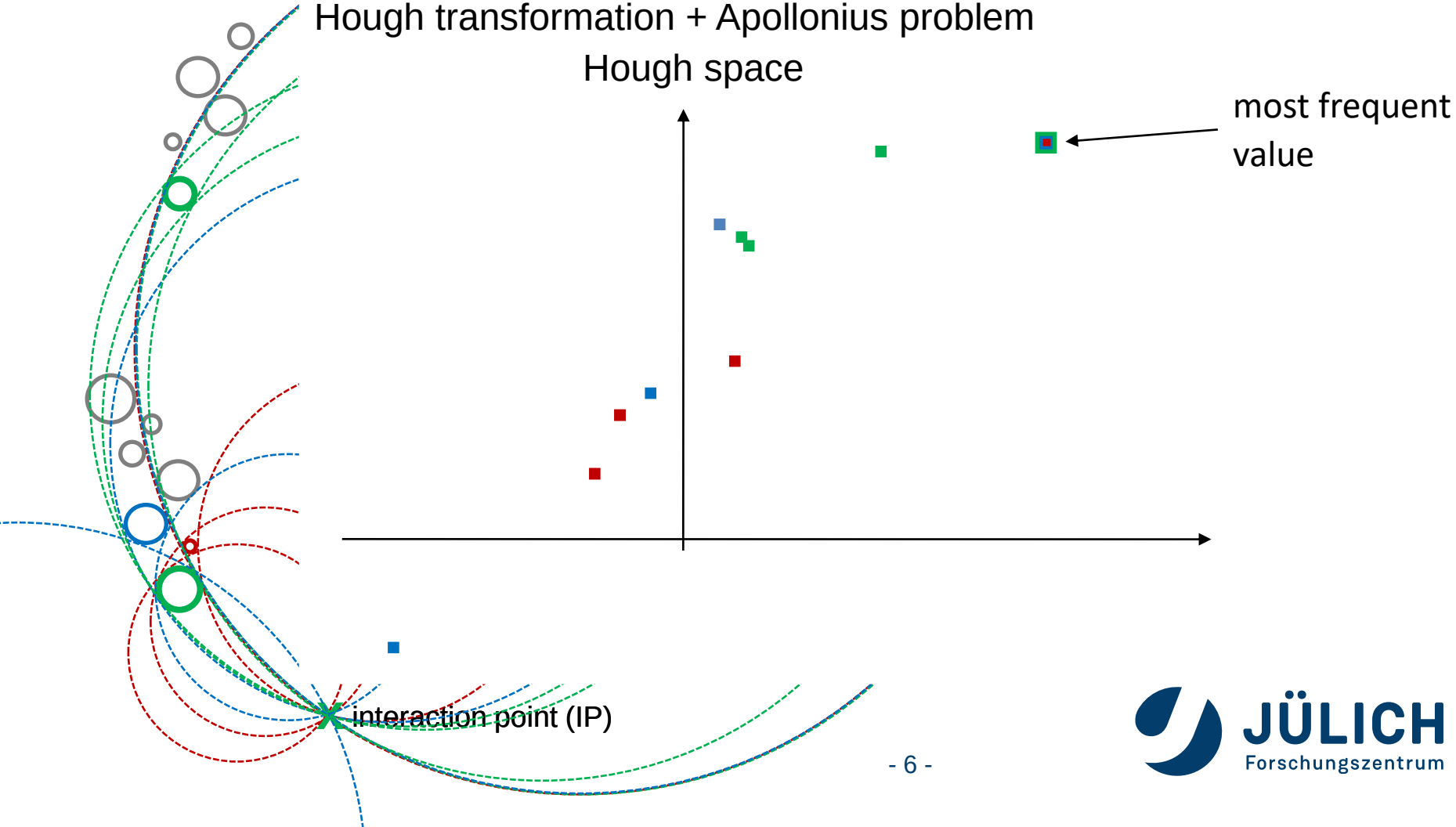
Hough transformation + Apollonius problem



WHY USING GPUS?

Does the HoughTrackFinder fulfill the conditions?

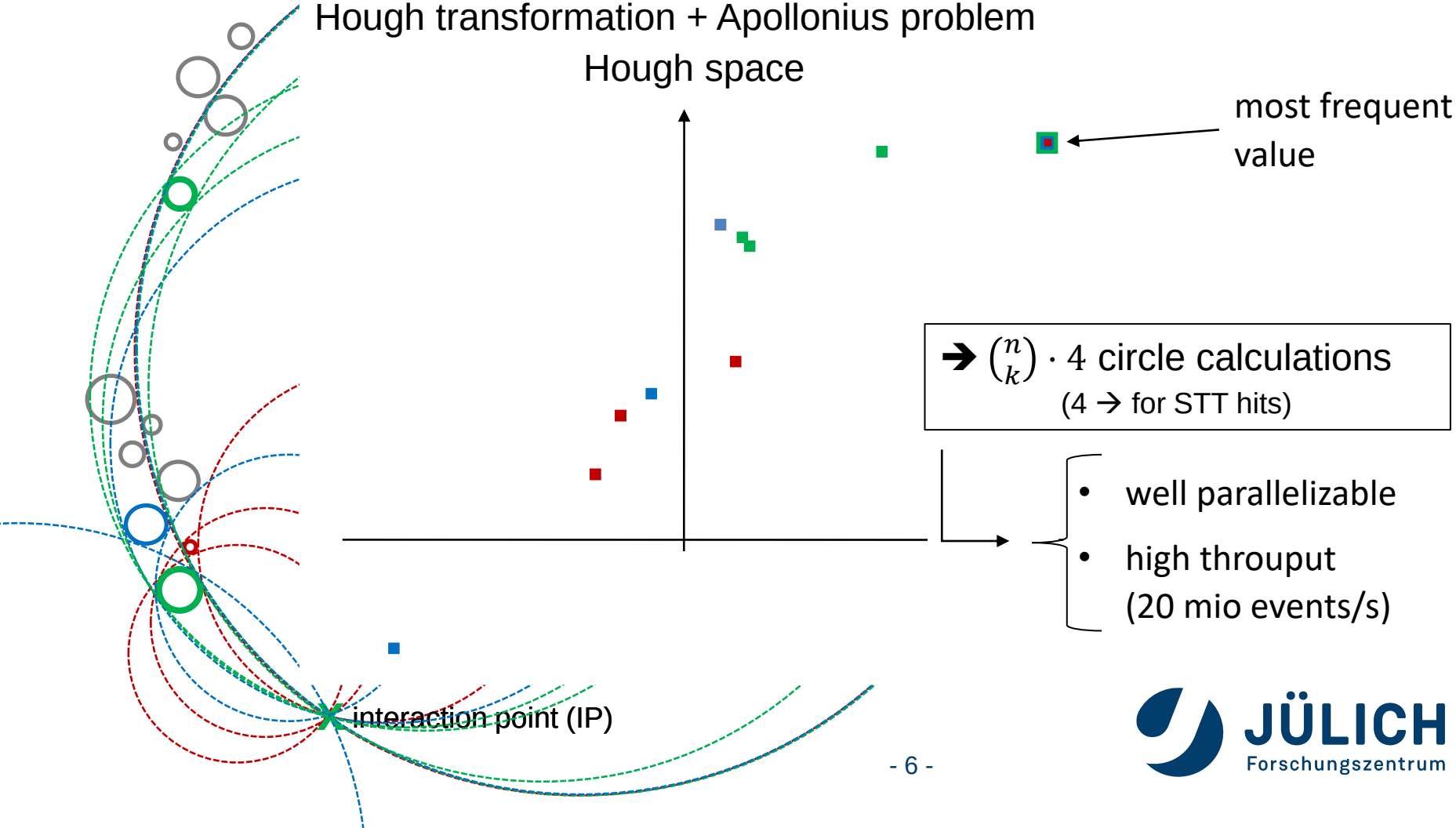
Hough transformation + Apollonius problem



WHY USING GPUS?

Does the HoughTrackFinder fulfill the conditions?

Hough transformation + Apollonius problem

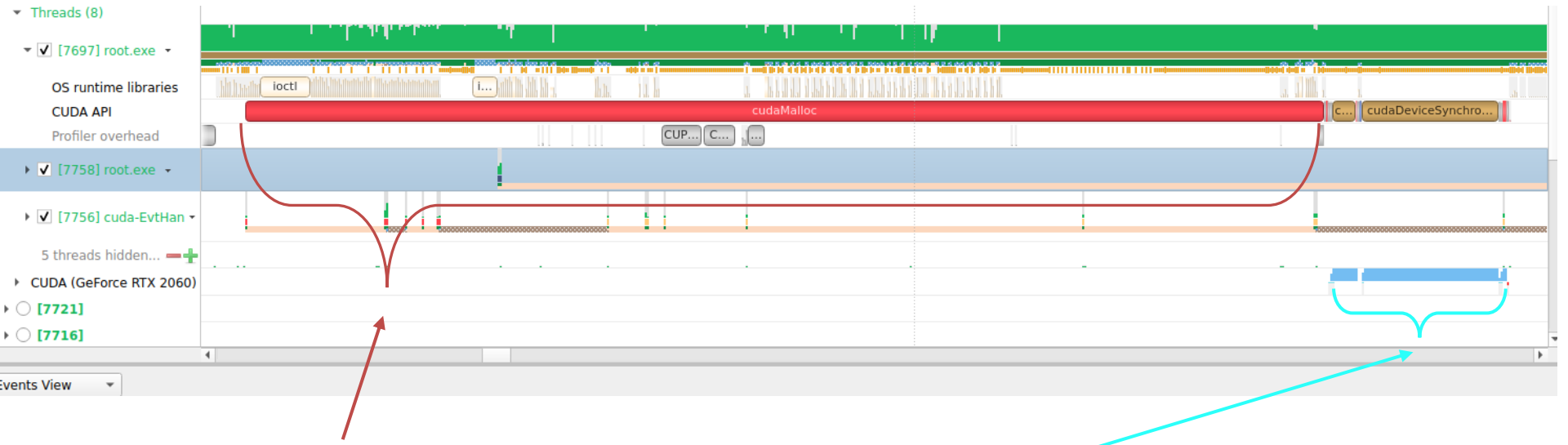


GPU PROGRAMMING CHALLENGES

- Allocate memory and copy data to GPU is time intensive

→ Main aims:

  - Reduce amount of data that have to be copied or allocated
  - Increase computing time of GPU
  - Increase occupancy of GPU



Allocate memory on GPU

Computing time on GPU

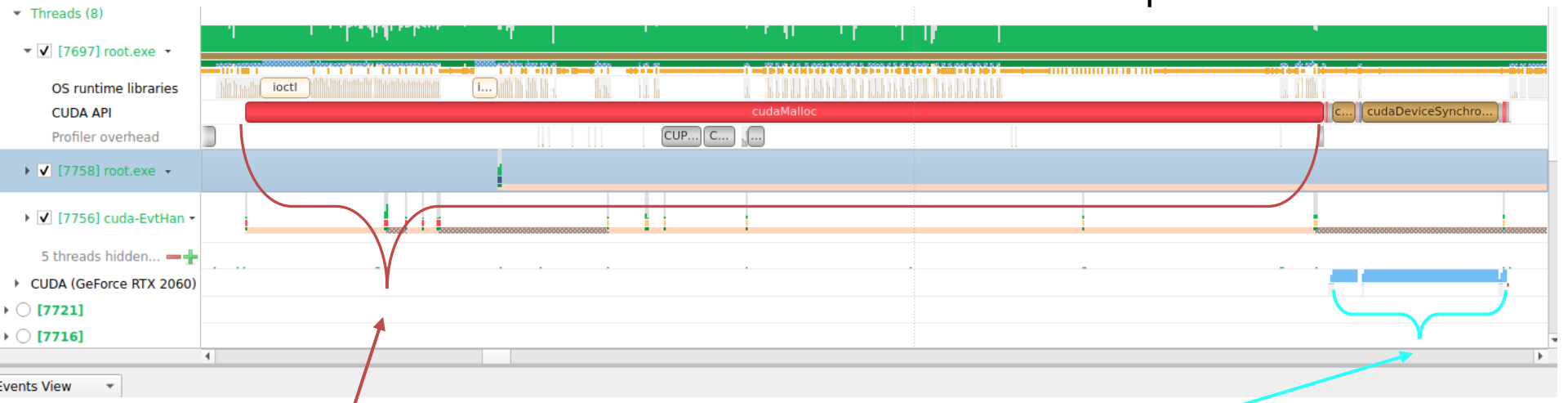
GPU PROGRAMMING CHALLENGES

- Allocate memory and copy data to GPU is time intensive

→ Main aims:

  - Reduce amount of data that have to be copied or allocated
  - Increase computing time of GPU
  - Increase occupancy of GPU

} Many events have to be calculated in parallel

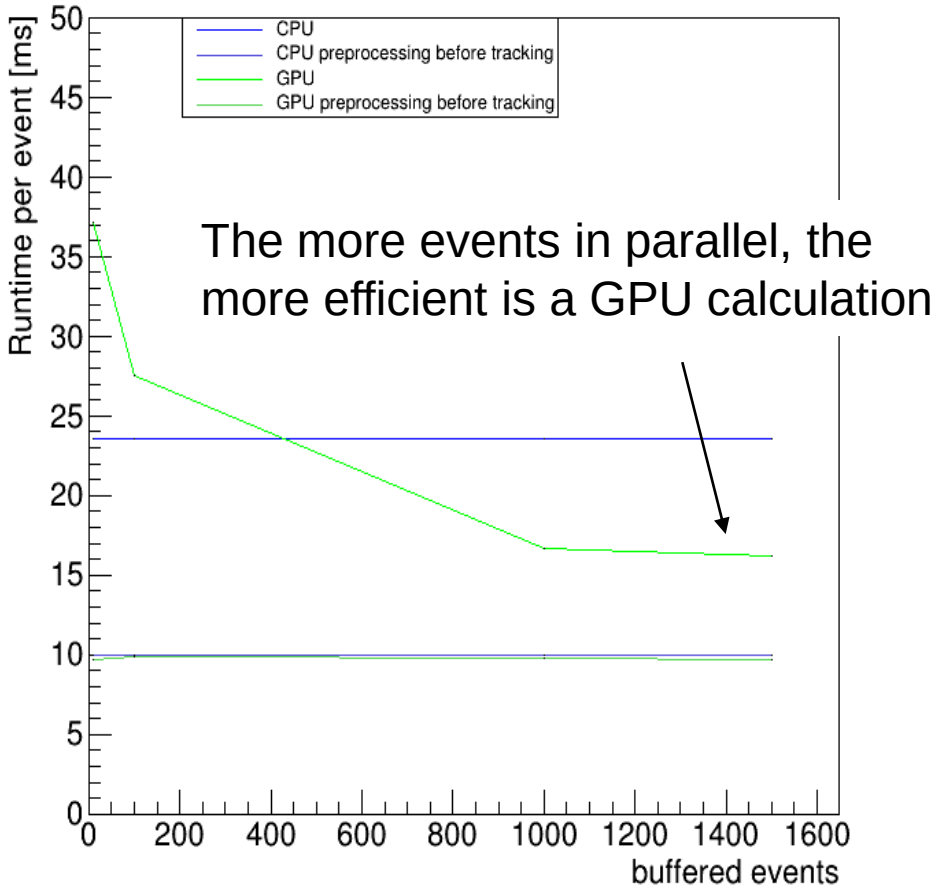


Allocate memory on GPU

Computing time on GPU

COMPARISON CPU VS GPU

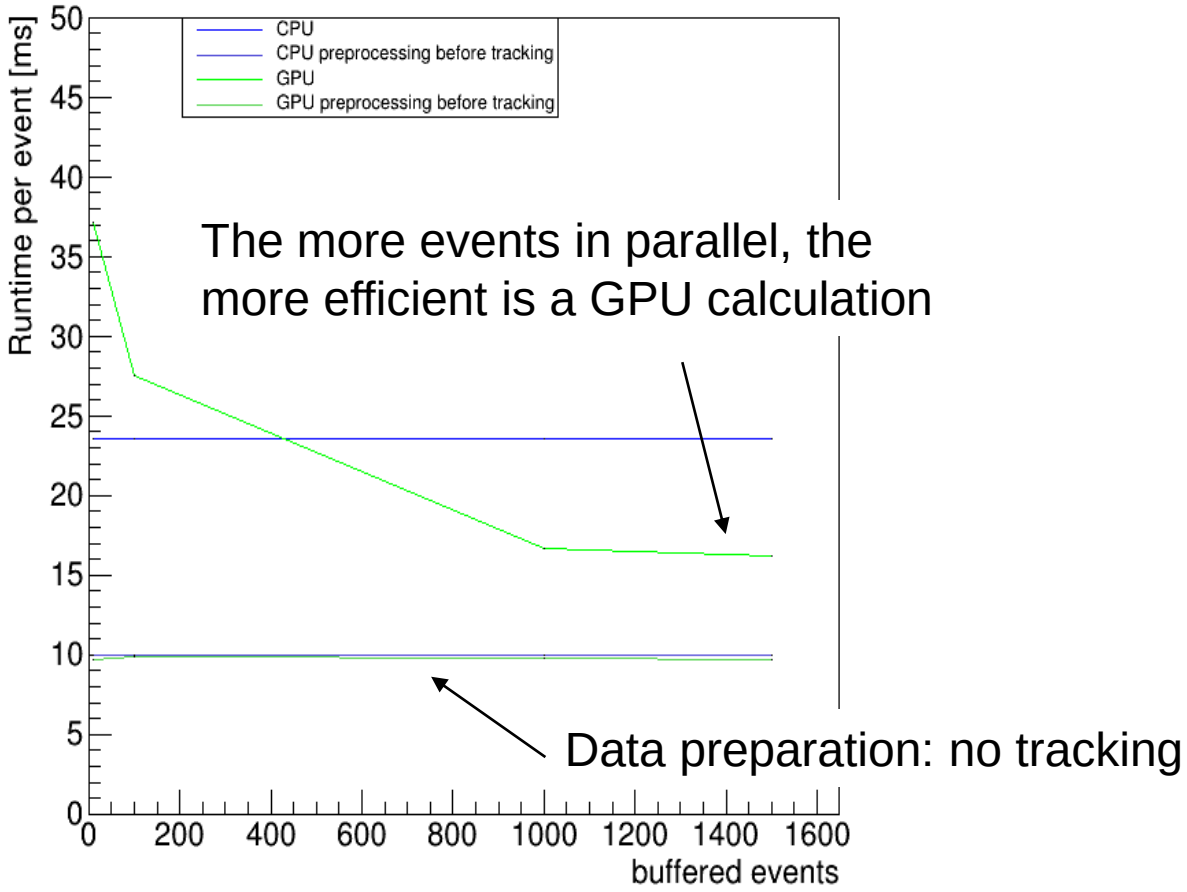
- Buffered events: events calculated in parallel



The more events in parallel, the more efficient is a GPU calculation

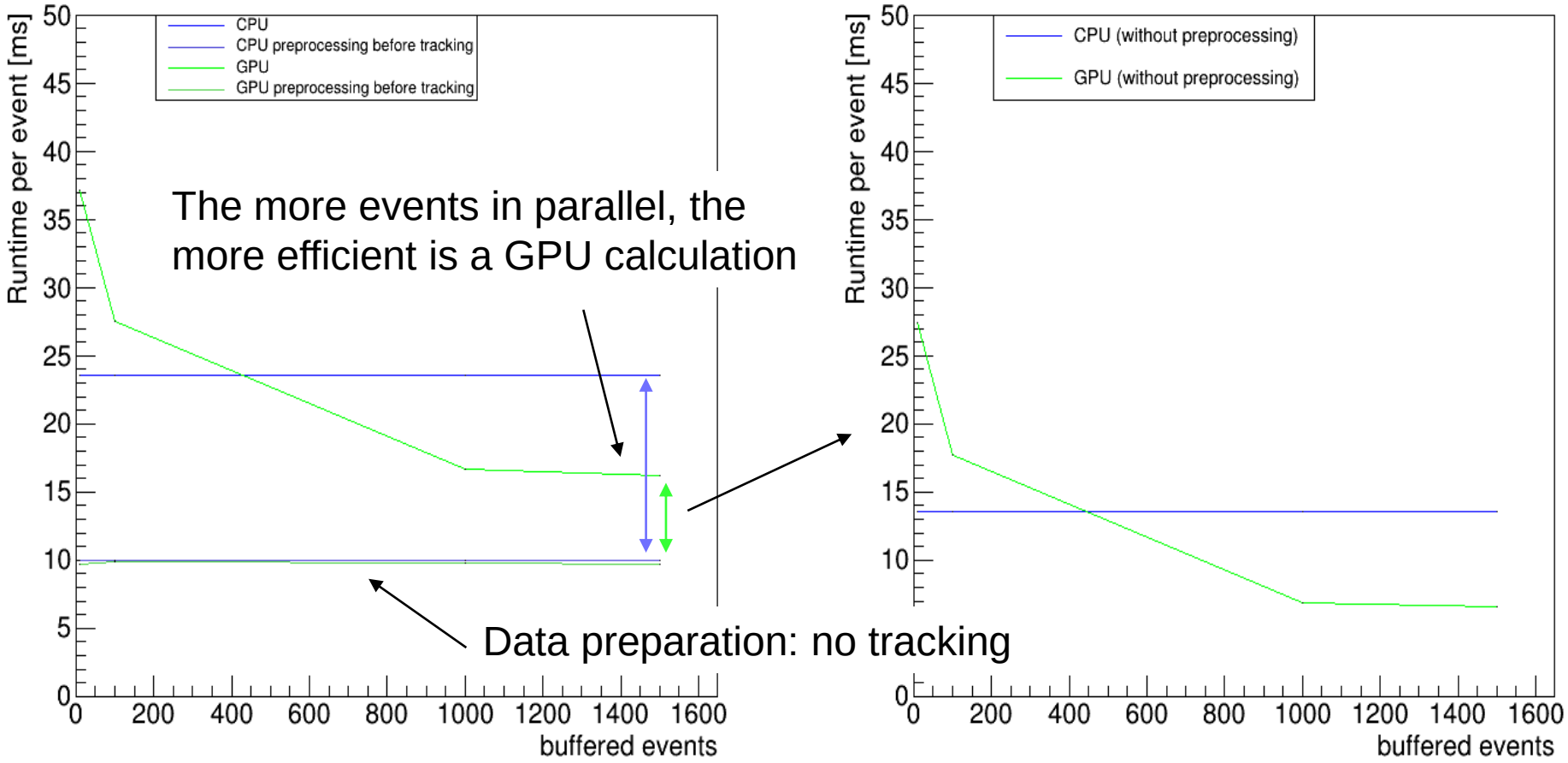
COMPARISON CPU VS GPU

- Buffered events: events calculated in parallel
- Before tracking data have to bee collected.
  → Software specific, different for real detector



COMPARISON CPU VS GPU

- Buffered events: events calculated in parallel
- Before tracking data have to bee collected.
  → Software specific, different for real detector



RESULTS

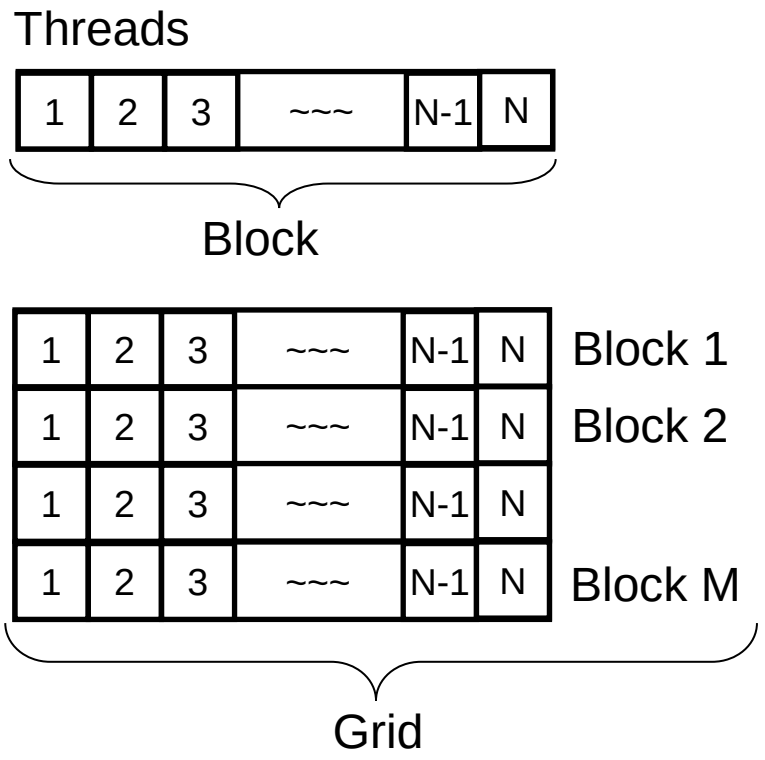
10,000 events, 7 GeV/c beam momentum, FTF background

	CPU time [ms/event]	GPU Time [ms/event]	Speed up
Apollonius calculation	4.25	0.05	85
Hough Space	7.27	2.35	3
Hough transformation (Maximum Finding)	8.37	3.19	2.6
With merging	12.54	4.88	2.5

RESULTS

10,000 events, 7 GeV/c beam momentum, FTF background

	CPU time [ms/event]	GPU Time [ms/event]	Speed up
Apollonius calculation	4.25	0.05	85
Hough Space	7.27	2.35	3
Hough transformation (Maximum Finding)	8.37	3.19	2.6
With merging	12.54	4.88	2.5

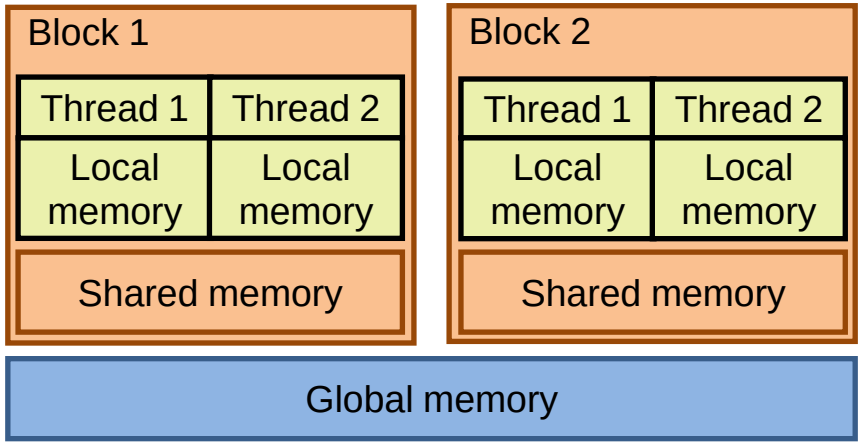


RESULTS

10,000 events, 7 GeV/c beam momentum, FTF background

	CPU time [ms/event]	GPU Time [ms/event]	Speed up
Apollonius calculation	4.25	0.05	85
Hough Space	7.27	2.35	3
Hough transformation (Maximum Finding)	8.37	3.19	2.6
With merging	12.54	4.88	2.5

Fast thread memory is too small
→ Slow global memory accesses



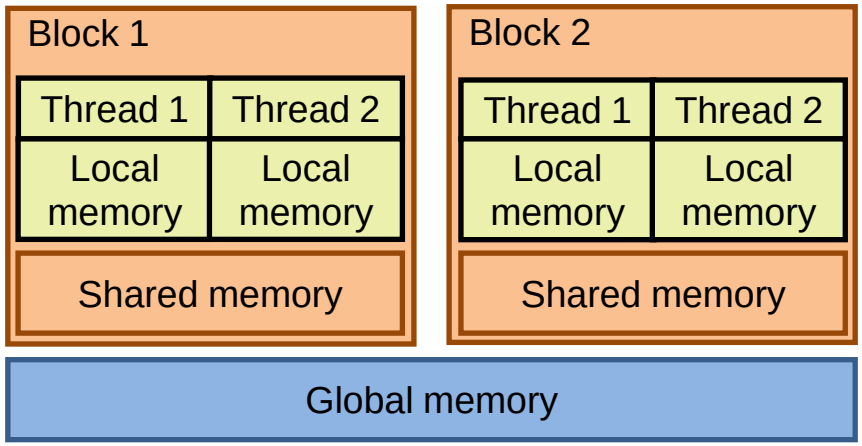
RESULTS

10,000 events, 7 GeV/c beam momentum, FTF background

	CPU time [ms/event]	GPU Time [ms/event]	Speed up
Apollonius calculation	4.25	0.05	85
Hough Space	7.27	2.35	3
Hough transformation (Maximum Finding)	8.37	3.19	2.6
With merging	12.54	4.88	2.5

A whole method per thread

Fast thread memory is too small
→ Slow global memory accesses



1 Apollonius Circle per thread

RESULTS

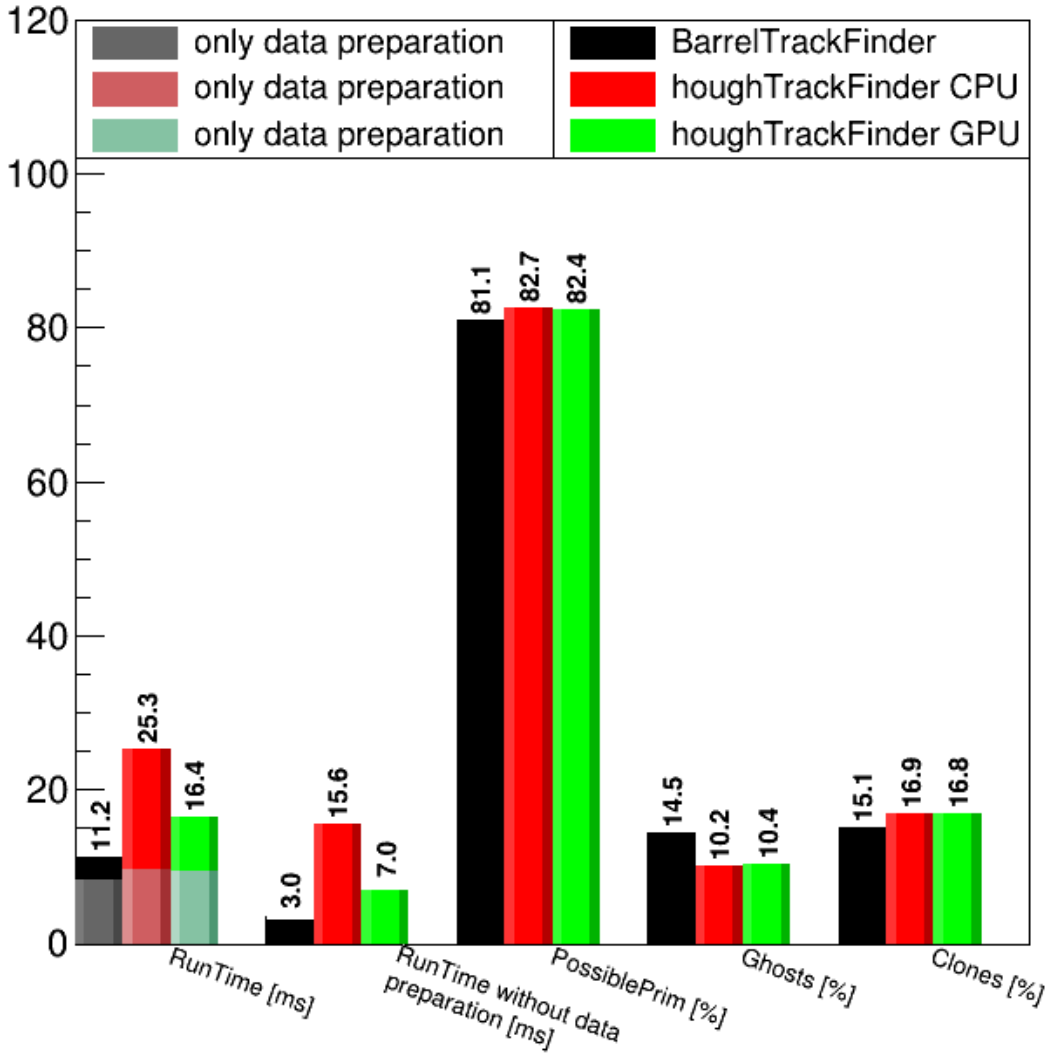
10,000 events, 7 GeV/c beam momentum, FTF background

	CPU time [ms/event]	GPU Time [ms/event]	Speed up	Memory usage [GB]	Buffer Size	Blocks * Threads
Apollonius calculation	4.25	0.05	85	2	10000	393812 * 256
Hough Space	7.27	2.35	3	4	4000	125 * 256
Hough transformation (Maximum Finding)	8.37	3.19	2.6	4	4000	125 * 256
With merging	12.54	4.88	2.5	4	2000	8 * 256

→ Bad GPU occupancy

→ Potential for improvement

QUALITY + RUNTIME COMPARISON



- GPU version is a bit faster
- Still too slow
- Potential for development

SUMMARY & OUTLOOK

Summary

- GPU version of HoughTrackFinder is implemented
- → Large speed up only for one part of the algorithm
- Speed up in total: ~2

Outlook

- Speed: Potential for development
  - → memory problem: allocate all Hough spaces for all tracklets
  - → better method has to be found
- Secondaries: as new improvement for tracking in PANDA

Thank you for your attention!