

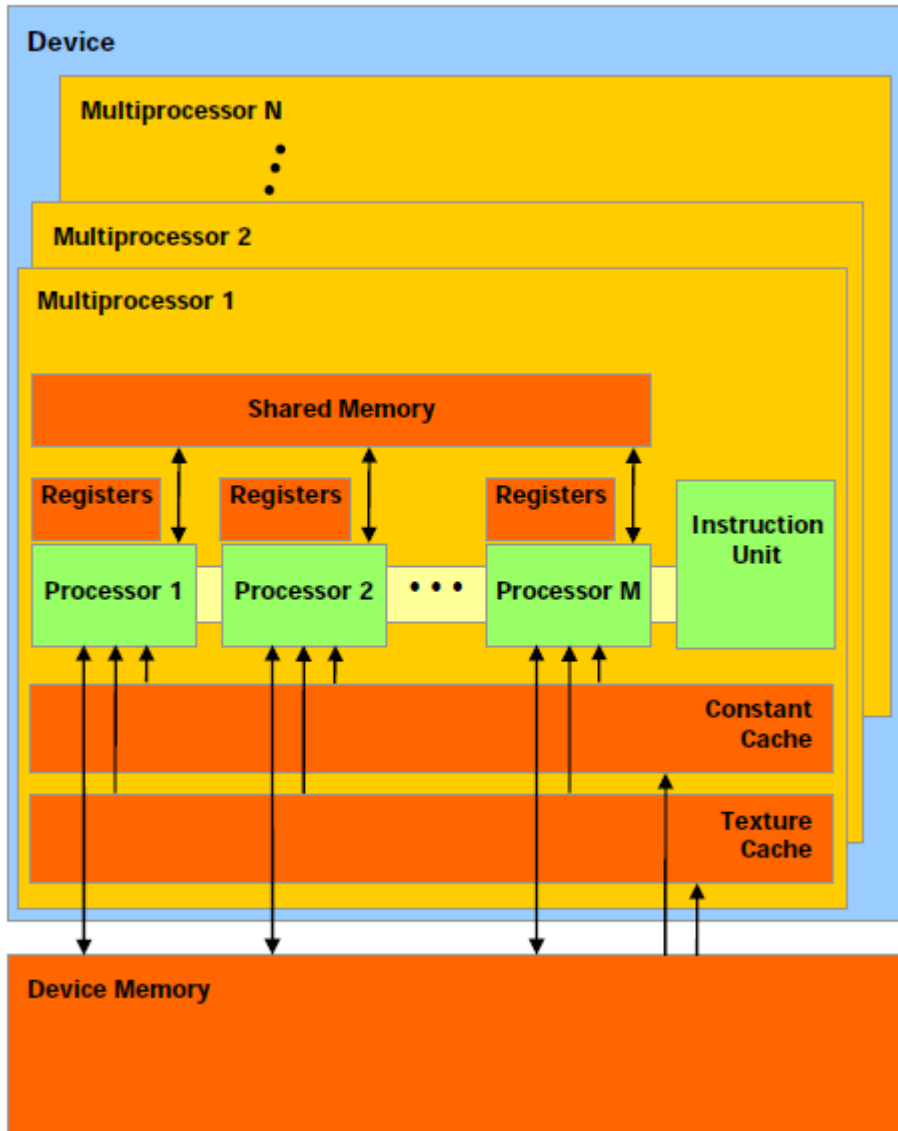
Alice TPC Online Tracker for GPU

David Rohr

Kirchhoff Institute for Physics, University of Heidelberg

September 7, 2009

Nvidia GPU Overview



Nvidia Geforce GTX285:

- 30 independant Multiprocessors
- 1 GB DDR device memory (no cache)
- 65536 bytes constant memory (cached)

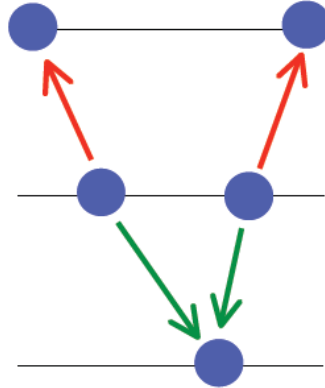
Multiprocessor Capabilities:

- 8 parallel ALUs (single precision)
- 16384 bytes shared memory
- 16384 32-bit registers
- highly pipelined
- about 256 threads recommended
- organized in warps of 32 threads
- single instruction per warp

Tracker overview

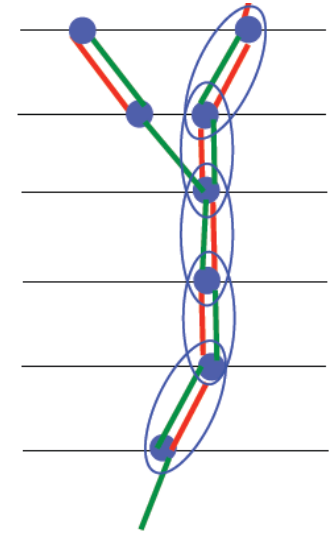
1. Neighbours finder

- For each TPC cluster find two (up&down) neighbours which compose the best line



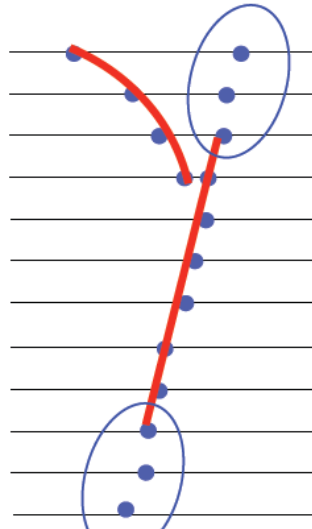
2. Composing of tracklets

- One-to-one linked neighbours are grouped to the track segments



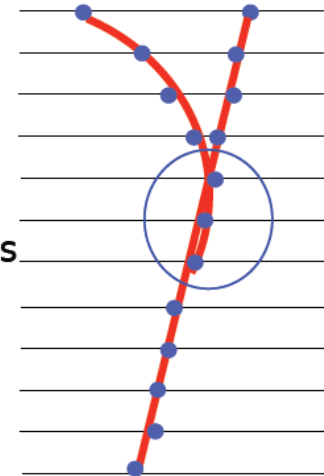
3. Construction of tracklets

- Fit of trajectories
- Search for the missed parts



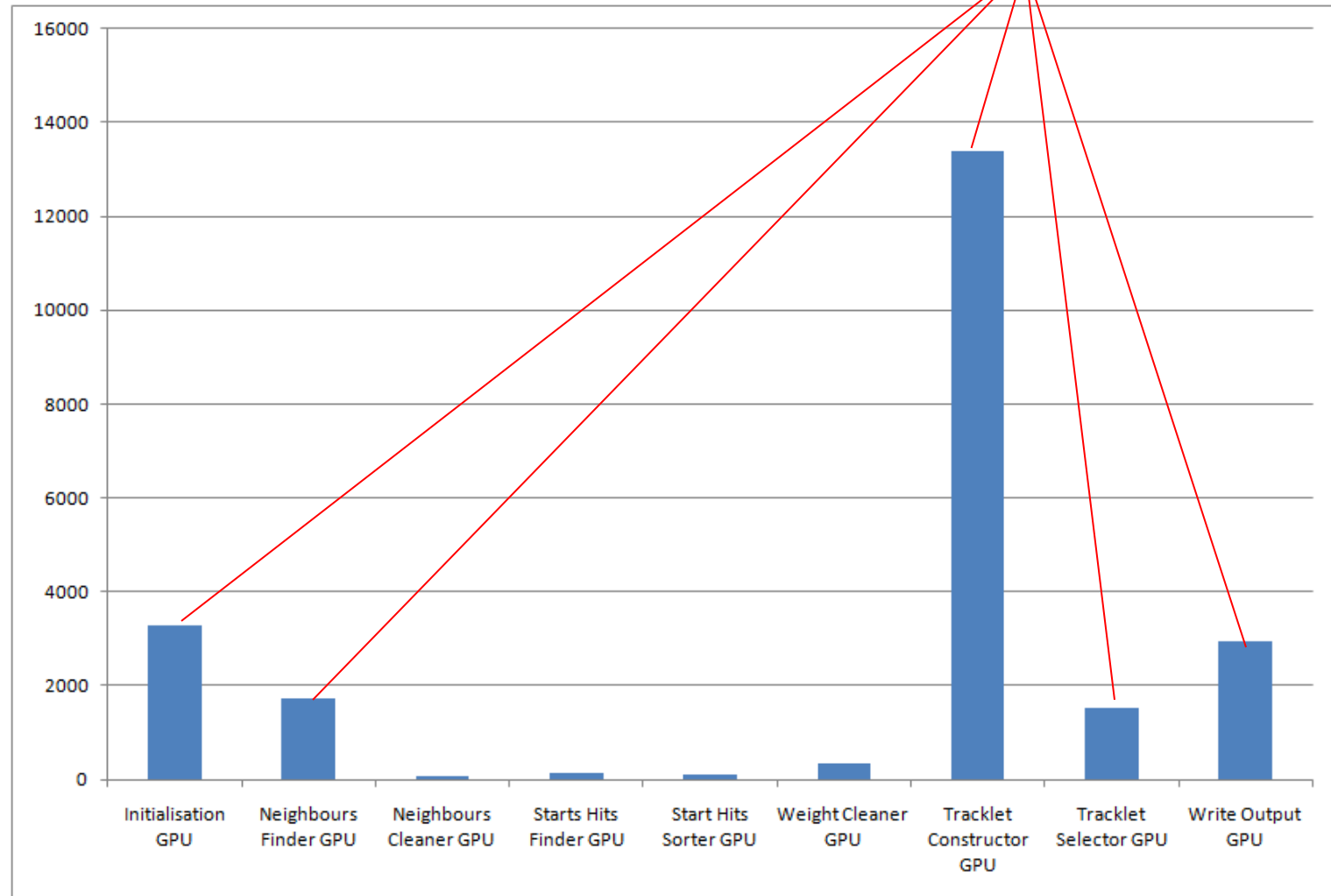
4. Final selection of tracks

- Competition between tracks, few shared clusters are allowed



Time Consumption of Tasks

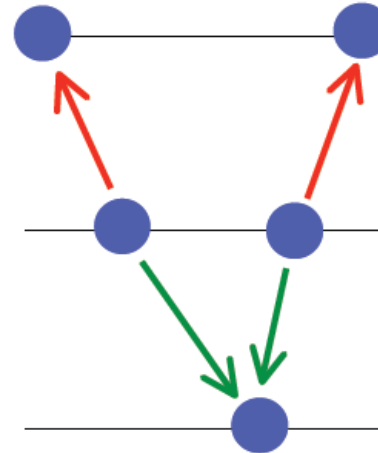
Relevant Tasks:



Distribution of Tracking Tasks on GPU

1. Neighbours finder

- For each TPC cluster find two (up&down) neighbours which compose the best line

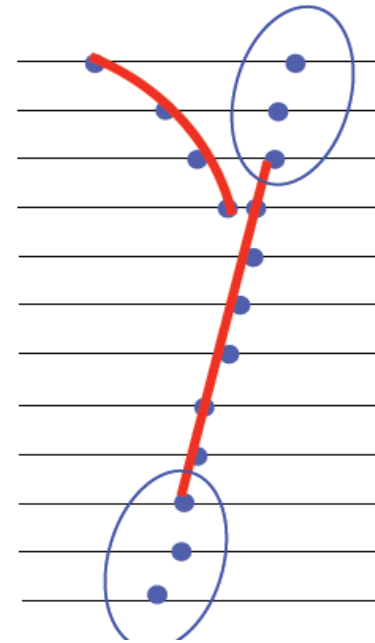


Task is entirely independant.
Distribute rows among Multiprocessors.
Each thread processes some hits in one row.

Distribution of Tracking Tasks on GPU

3. Construction of tracklets

- Fit of trajectories
- Search for the missed parts



Tracklets are independent.
Thus tracklets are distributed among threads.
Arbitrary distribution among multiprocessors.

Approx. 15k tracklets in central heavy ion collision.

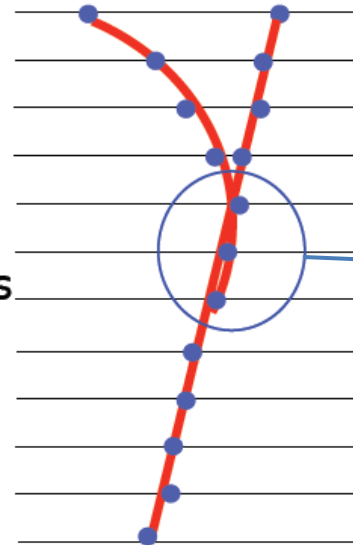
$256 * 30 = 7680$ threads recommended.

Approx 2 tracklets per thread, probably suboptimal for tracklets with different length

Distribution of Tracking Tasks on GPU

4. Final selection of tracks

■ Competition between tracks, few shared clusters are allowed



Which tracklet do this hits belong to?

Tracklets are not independent.
Nonetheless use one thread per tracklet.
Coordination between Multiprocessors required.

GPU Memory Usage

Available Memory

Cached Constant Memory: 64 Kb

Shared Memory: 16 Kb

Global Memory: 1 Gb

Required Memory

Global Tracker Parameters (ca 2 Kb)

Row Parameters (ca 13 Kb)

Hits/Links in one Row, central pb-pb (ca 20 Kb)

Hits/Links in one Row, pb-pb (ca 7 Kb)

Total Memory (ca 60 Mb)

Scenarios:

1 Slice, non central pb-pb:

All Parameters fit in Constant Memory, Hits / Links per row can be cached in Shared Memory.

Up to 4 Slices, central pb-pb:

All Parameters fit in Constant Memory, Hits / Links do not fit in Shared Memory.

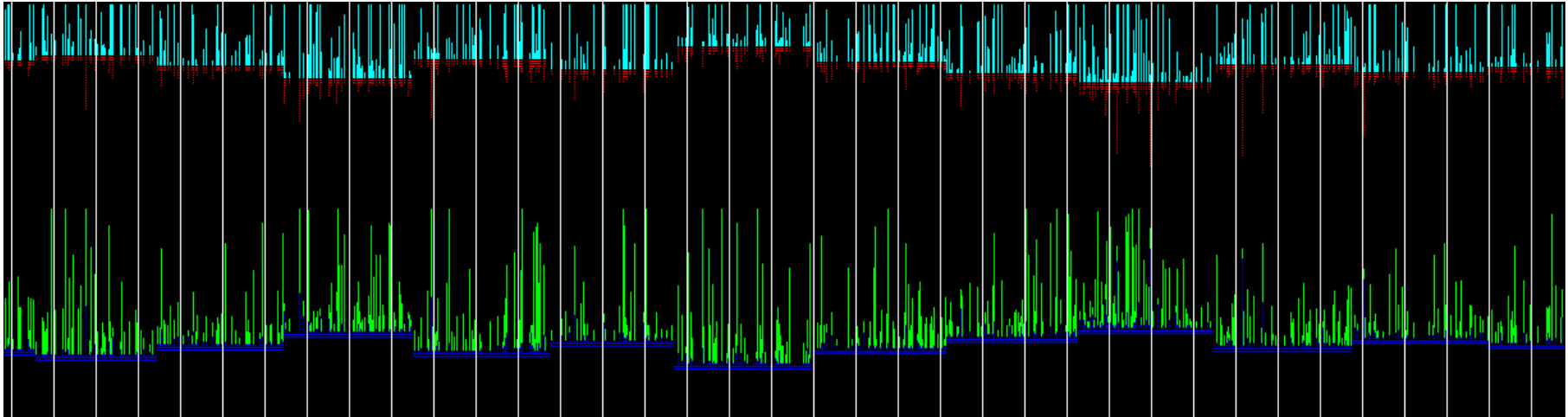
Up to 15 Slices, central pb-pb:

Only global Parameters fit in Constant Memory, Row parameters fit in Shared Memory.

More than 15 Slices:

Global Memory insufficient.

Tracklet Constructor Thread Utilization



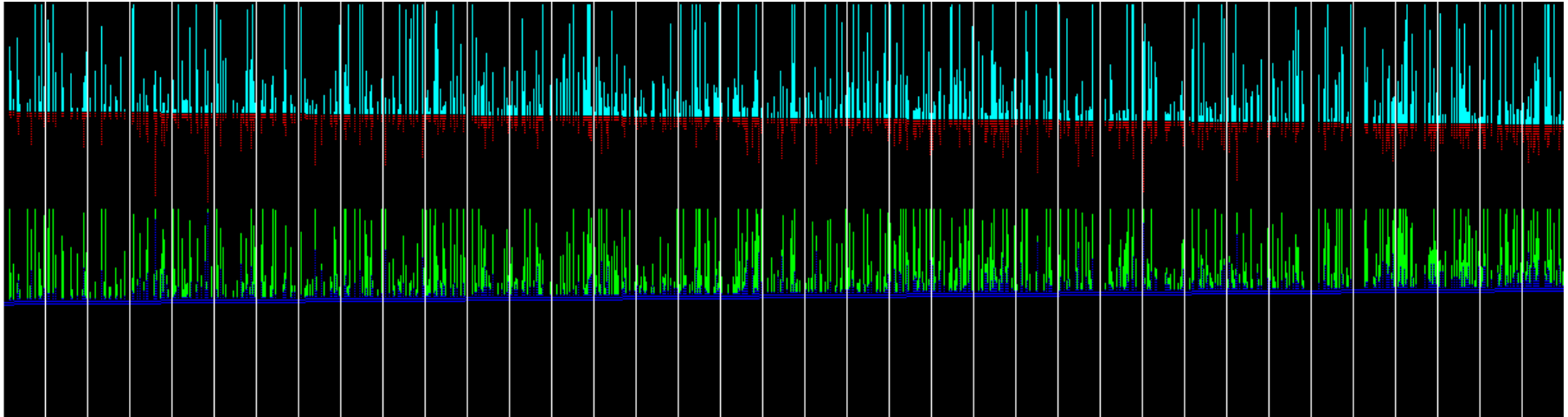
Threads on x-axis, Time on y-axis.

Threads process tracklets row by row, synchronous row in one warp.

Black:	Thread inactive
Blue:	Thread fitting
Green:	Thread extrapolating up
Red / light blue:	Thread extrapolating down
White:	Border between warps

***Bad Multiprocessor
utilization***

Tracklet Constructor Thread Utilization

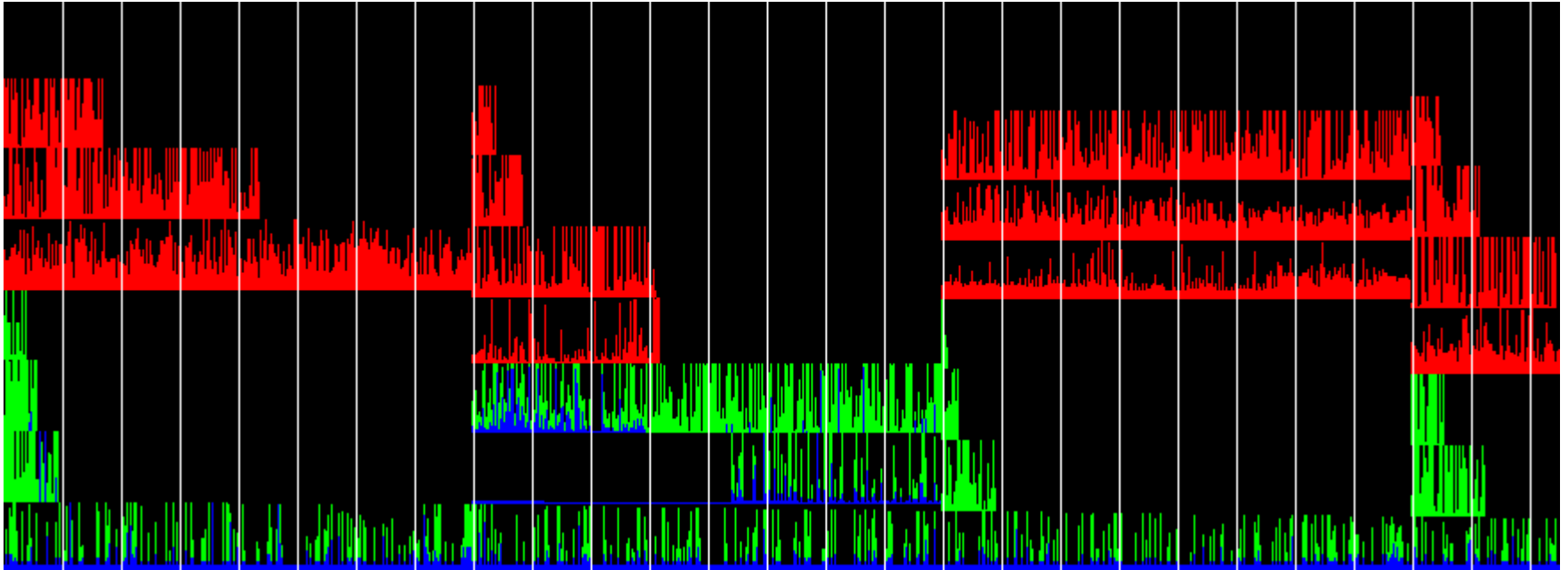


Sorting start hits to reduce divergent branches in warp gave about 3% performance increase.

Biggest problem is different length of tracklets.
All threads in one warp wait, till longest tracklet is finished.

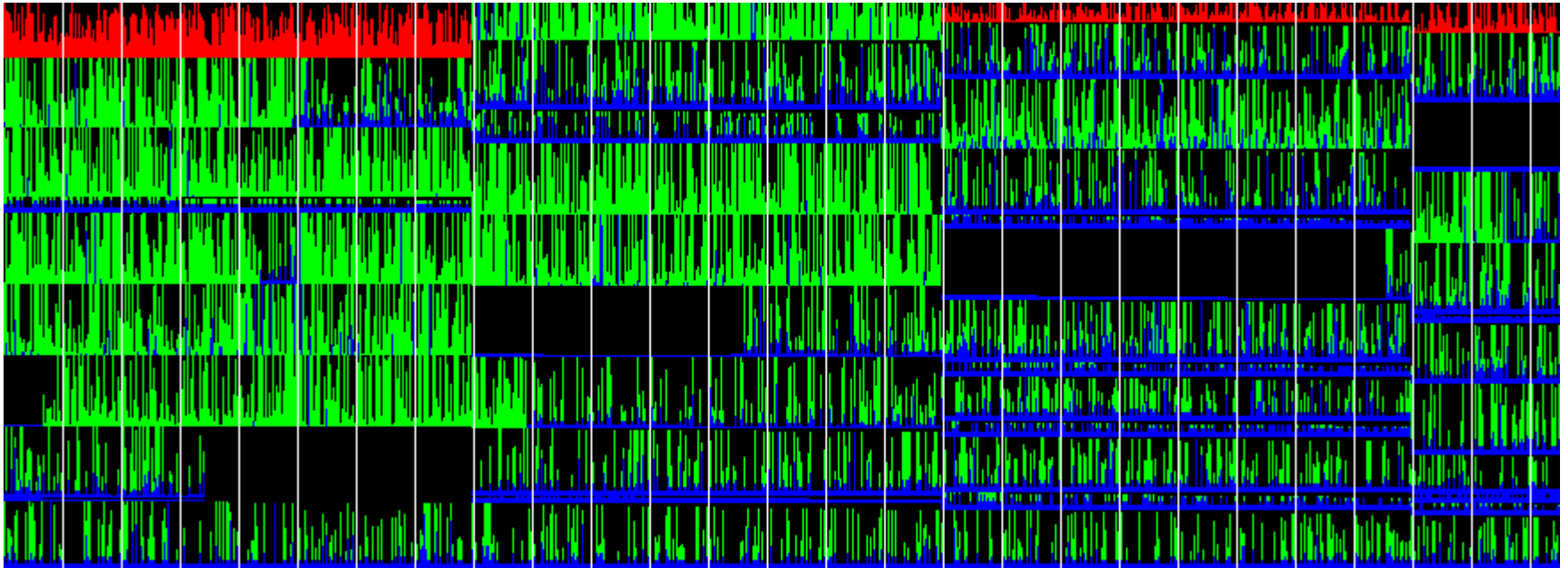
Better scheduling required for better performance.

Tracklet Constructor with scheduling



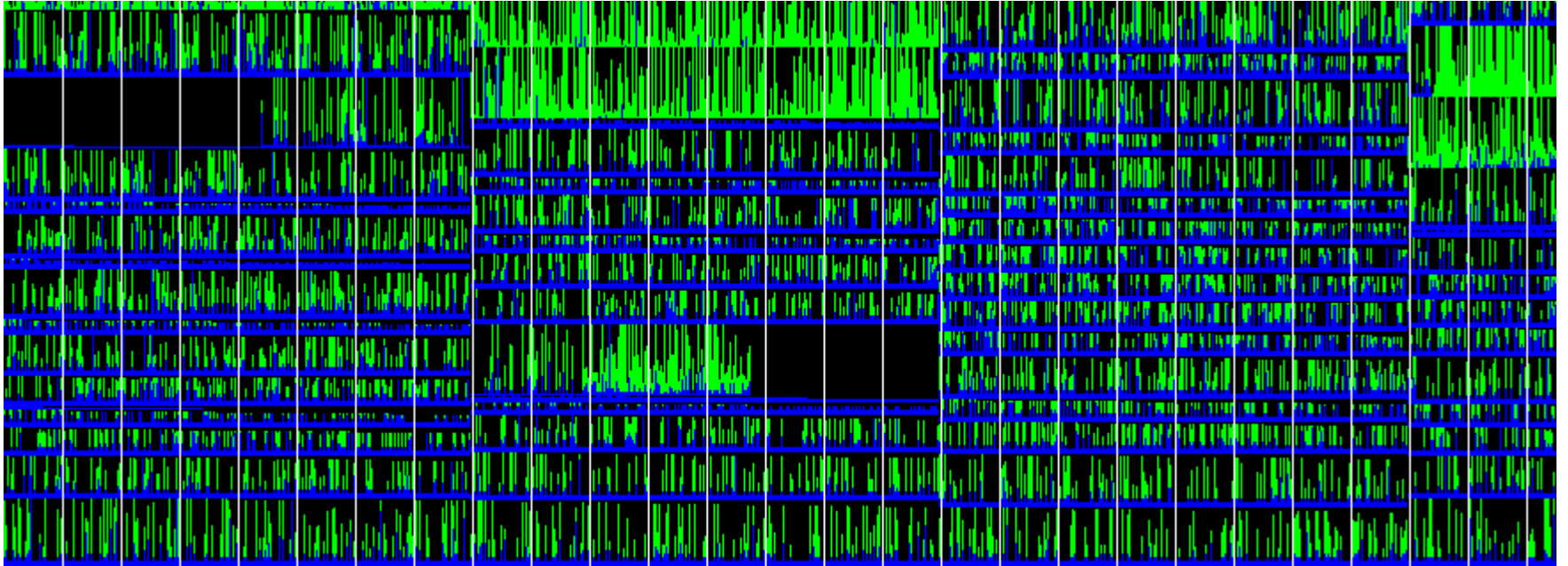
Redistributing tracklets after N rows results in better utilization of Warps.
Still too few tracklets to saturate GPU

Tracklet Constructor with scheduling



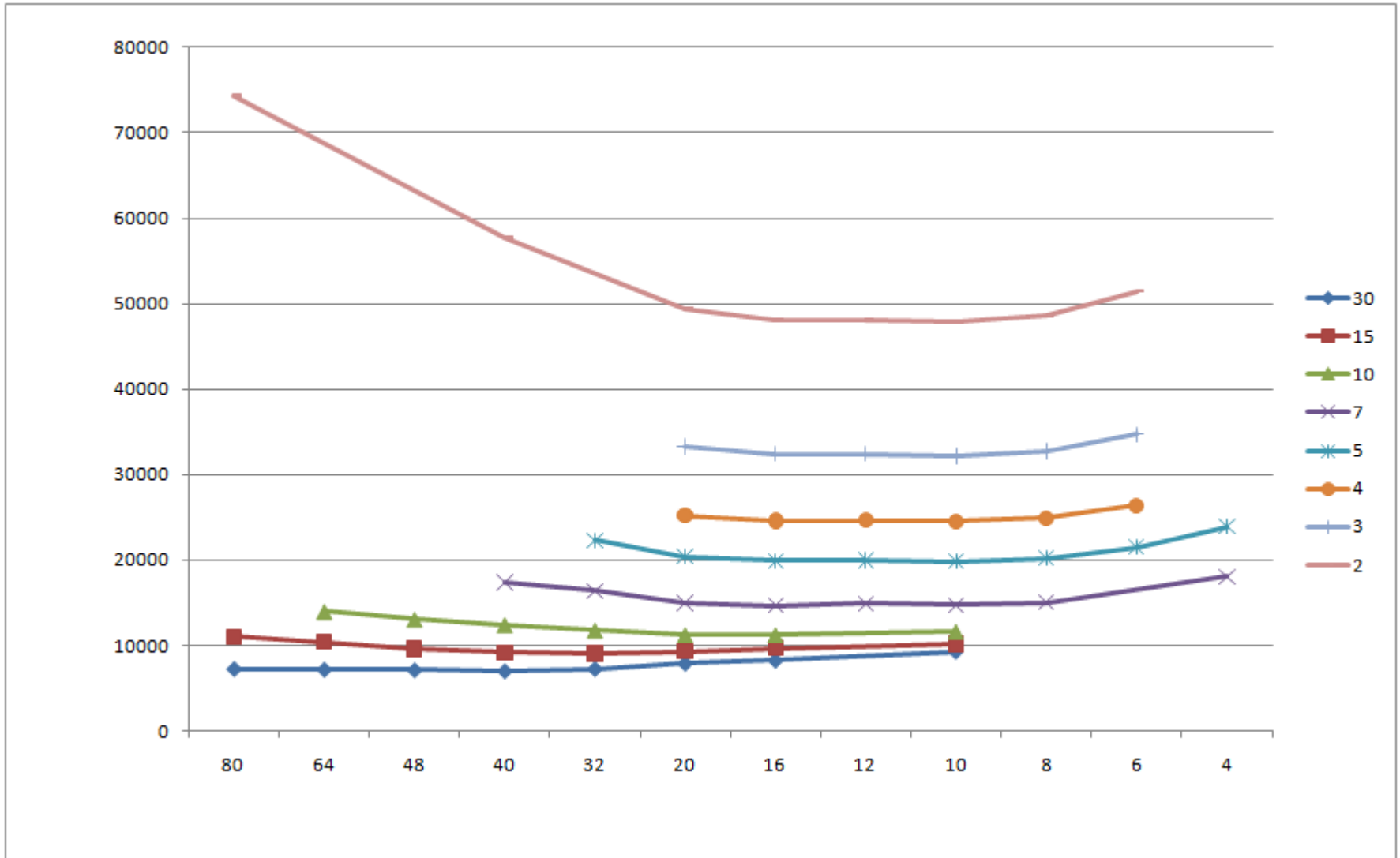
Scheduling while calculating 4 slices in parallel.

Tracklet Constructor with scheduling



Scheduling while calculating 9 slices in parallel.

Tracklet Constructor Benchmarks for different amount of Multiprocessors used (1 Slice).

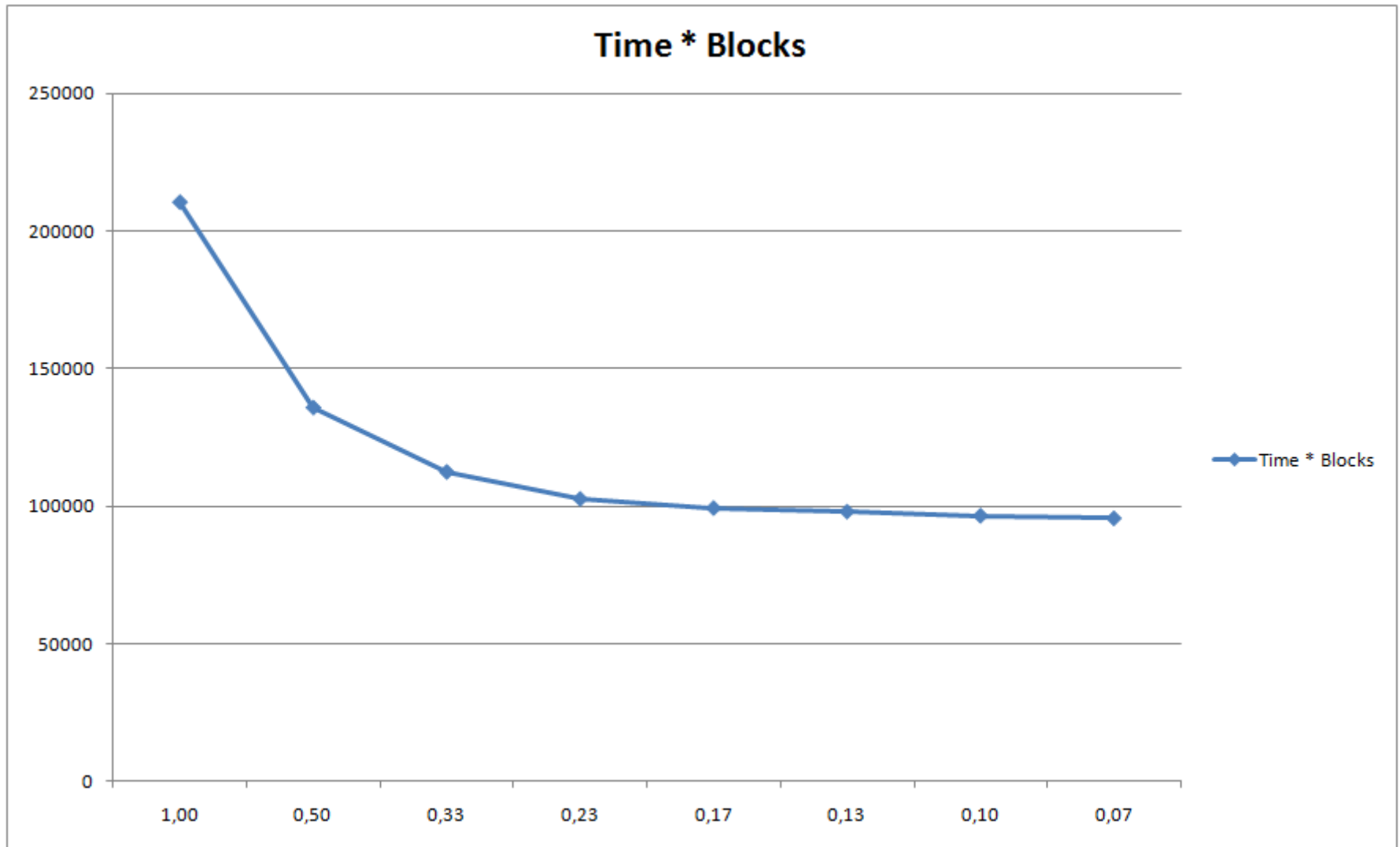


X: Amount of rows that are calculated between rescheduling.
Y: Calculation time.

Tracklet Constructor Benchmark Conclusions

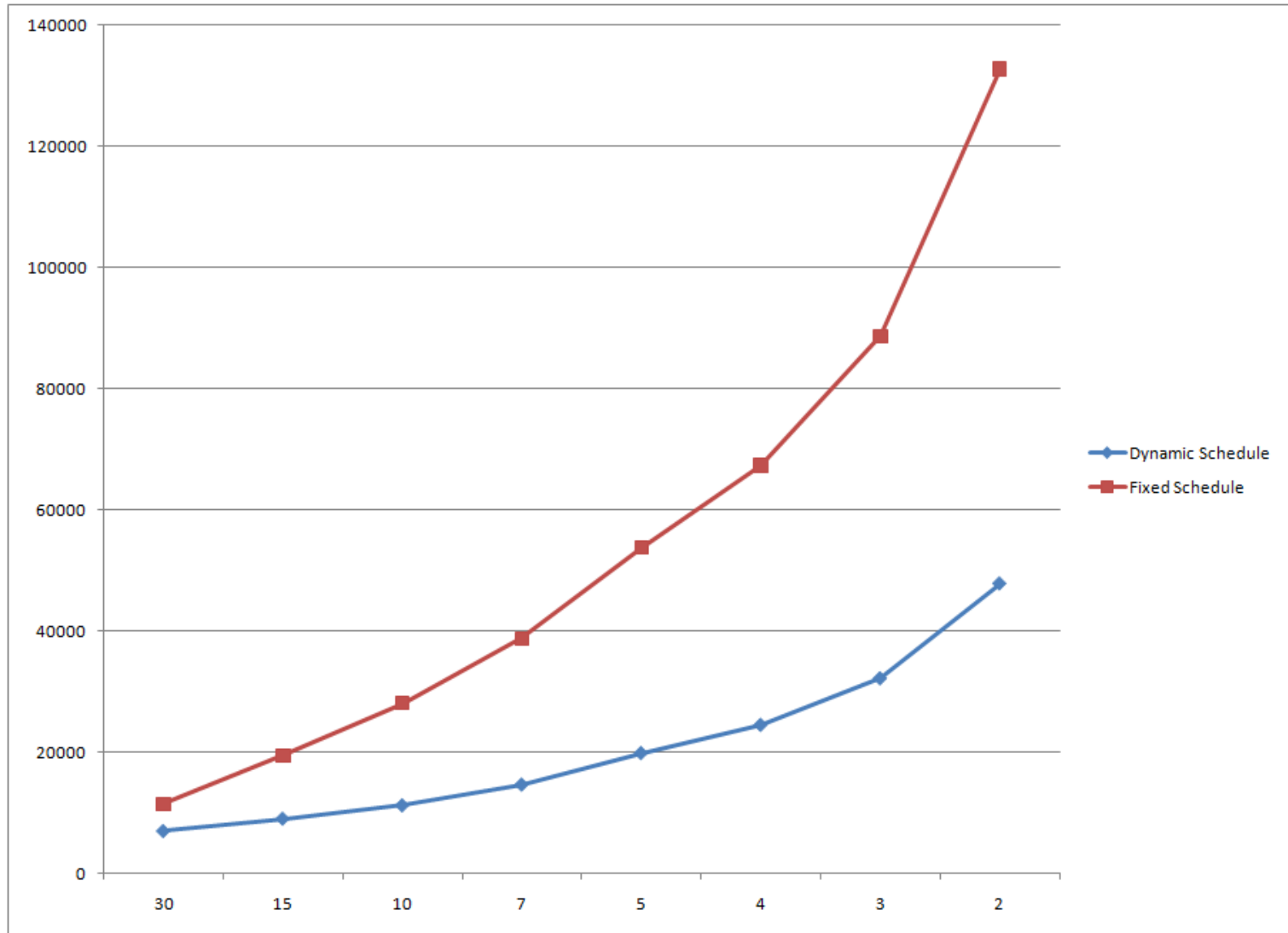
- Performance impact for 3 of 30 Multiprocessors used is a factor of 4 instead of 10.
- Optimal number of rows to calculate between rescheduling depends on device utilization.
- Calculating a single slice cannot saturate the GPU

Tracklet Constructor Benchmark Analysis



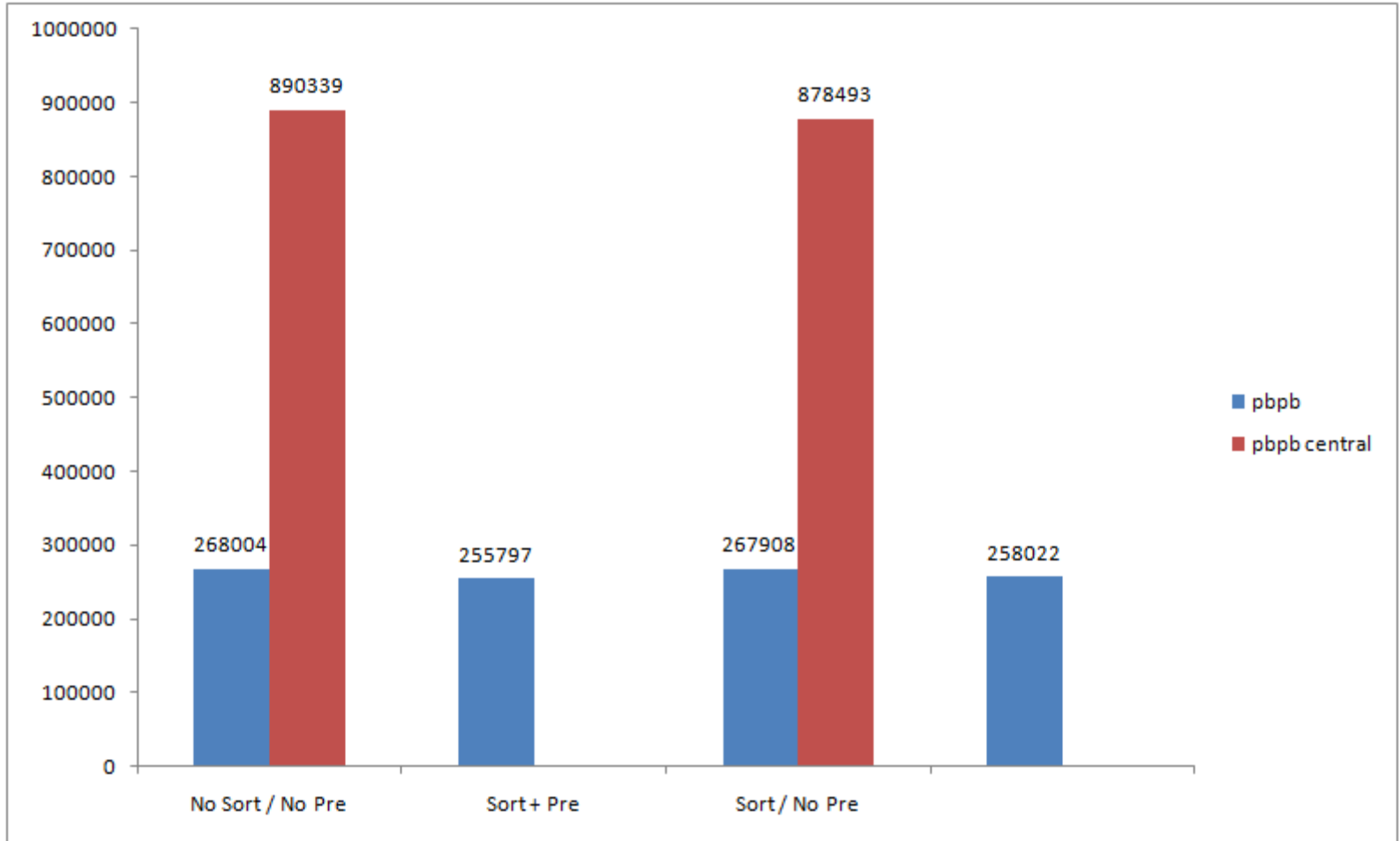
Calculation time normalized to 100% device utilization with real utilization on x-axis.
At least 4 slices should be calculated in parallel.

Tracklet Constructor Benchmark Analysis



Calculation Time for fixed and dynamic scheduling using different Multiprocessor counts. Better relative performance for less device utilization due to memory limitations.

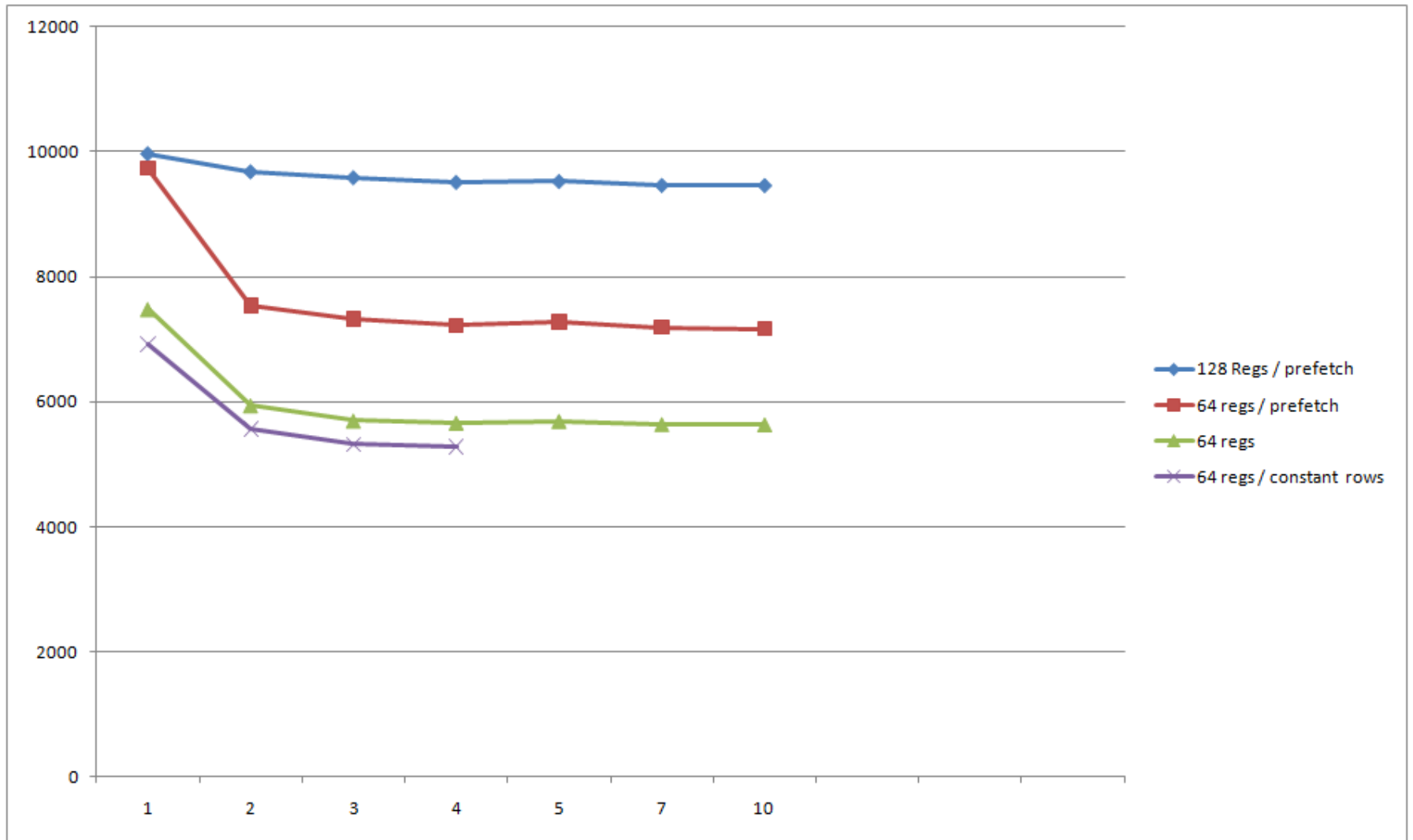
Tracklet Constructor Shared Cache Analysis



Prefetching not available for central pb-pb.

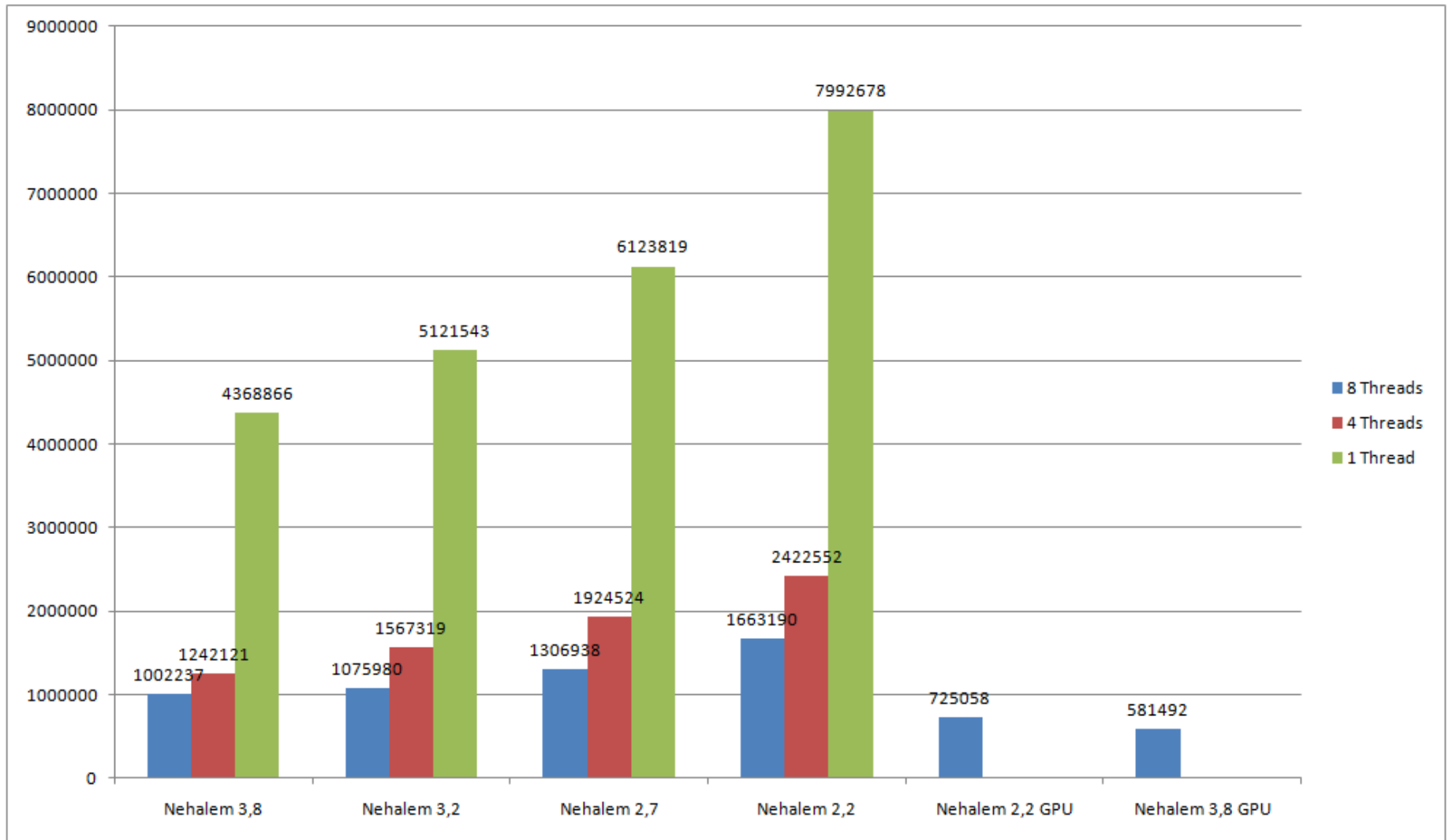
Prefetching gives 5% performance increase for non central.

Tracklet Constructor Benchmark Analysis

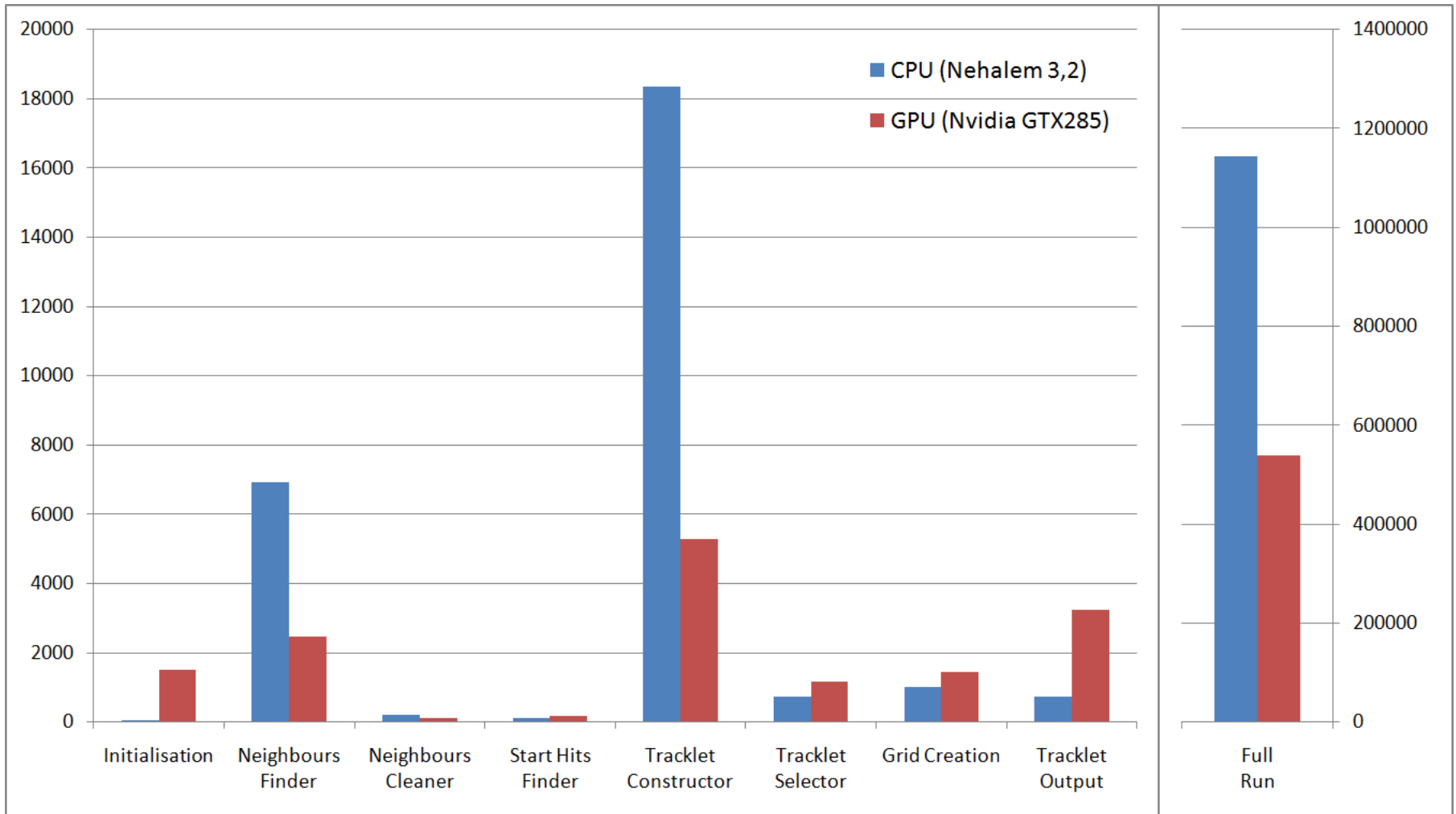


Partial Prefetching of Hits for central pb-pb. Requires local memory due to lack of registers. Using more registers reduces thread count and thus performance.

Tracklet Constructor Performance (August 2009)



Tracklet Constructor Performance (September 2009)



Performance Gain: 2,12