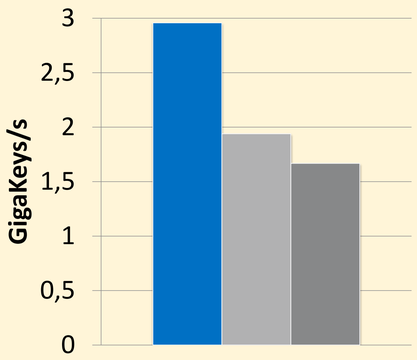
CONTACT NAME
Jimmy Pettersson: jimmy.pettersson@hpcsweden.se

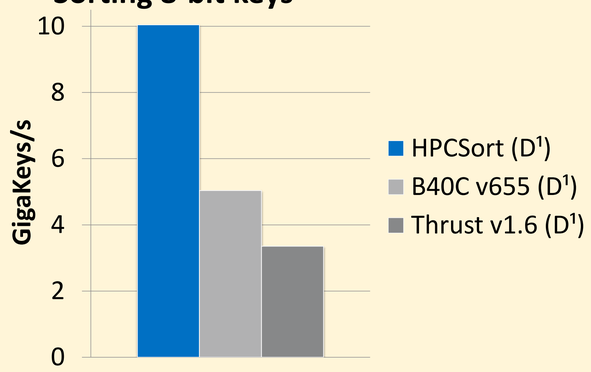
High performance sorting using efficient high-resolution histogram generation with CUDA

Jimmy Pettersson (HPC), Ian Wainwright (HPC)

Sorting 16-bit keys



Sorting 8-bit keys

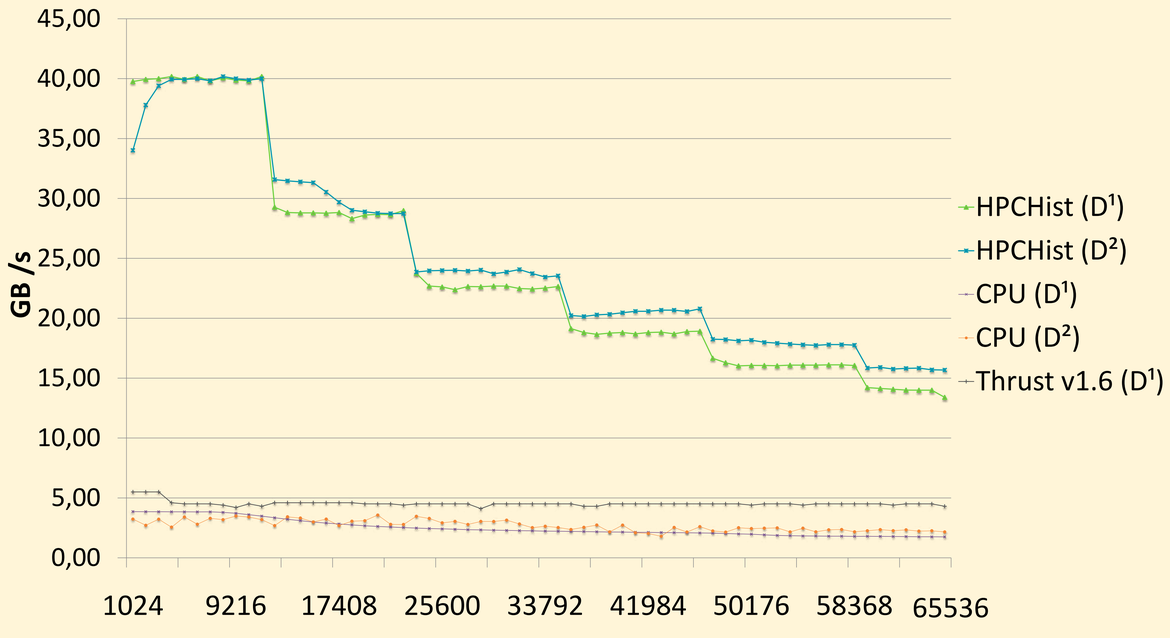


Application example: Real-time histogram equalization (HE) [500+ FPS @ 1920x1080:16-bit]

Before HE

After HE

Histogram performance over bin count



What it is

The fastest **sorting** implementation for 8-bit & 16-bit keys. This is implemented by efficiently generating **high-resolution histograms**.

Implementation

A large histogram is efficiently generated followed by a parallel prefix sum. The output from both the histogram and the prefix sum are then used to generate the sorted output array, essentially performing a counting sort.
The algorithm complexity is O(k+N), were we draw advantage of the fact that 'k' (the number of unique keys) is small compared to the number of elements, giving us a linear algorithm complexity.

Conclusions

- A **sorting speed-up of over 2x against the existing fastest GPU-based sorter** has been achieved.
- One can expect at least a **10-fold speedup** compared to contemporary CPUs for sorting applications.
- Real-time image enhancement and statistical analysis applications can draw great benefit in the capability to generate high-resolution histograms (65536+ bins) efficiently.

Datasets	
Element count	64 M Elements (2^{26})
D^1	Random uniform distribution
D^2	Random gaussian distribution
Key types	8-bit / 16-bit keys
Value type	32-bit values

GPU Implementations compared	
Thrust v1.6	Thrust
B40C v655	Back 40 computing
HPCSort	Our implementation

GPU	**GTX480**
Core count	480
Theoretical compute power	1344 GFLOPS
SDK Bandwidth test	146.51 GB/s
Non-GPU hardware	
Intel Core i7 960 3.2 Ghz 8 Logical Cores	
ASUS P6T Deluxe V2	
12 GB RAM	
Software	
Windows 7 64-bit SP 1	
CUDA 306.97 driver	
CUDA toolkit 4.2	

Acknowledgements
Thanks to Mark Harris (Nvidia), Duane Merrill (University of Virginia), and Sean Baxter (Nvidia) for valuable feeback and early release information of the latest B40C sorting implementation.