Parallel Data-processing on GPGPU
Radim Vansa, Charles University in Prague, Faculty of Mathematics and Physics

Motivation
Modern graphic cards are no longer limited to image rendering and geometric calculations but also allow parallel processing of non-graphical data. In practice, these general-purpose GPUs can be used in database management systems as co-processors, accelerating certain time-consuming tasks.

Challenges
The execution model used on GPUs called SIMT is substantially different from the common one used on multi-core systems. In order to achieve best performance, the programmer must effectively use the system of caches and adhere optimal memory access patterns.

Objectives
We have studied two problems often solved in database systems:
• sorting, used for index creation, duplicities removal or grouping
• set intersection, basically the table join, with either sorted or unordered sets

OpenCL
OpenCL framework was used as an open standard for parallel programming of heterogeneous systems. This allowed us to write GPGPU programs portable across graphic cards from different vendors.

Results and Conclusion
We have implemented and benchmarked parallel versions of many algorithms, bringing an extensive comparison of various approaches to our problems. Significant speedup was achieved compared to CPU-based solutions - for example our mergesort implementation was up to 12.4x faster than std::sort and up to 3.1x faster than tbb::parallel_sort. Our GPU set intersection algorithm was also more than twice faster than optimized parallel CPU algorithm.