УДК 004.021:004.272.2:004.67

Krasnokutska I.V., PhD, Assoc. Prof.
Hatsenko D.O., undergraduate
Yuriy Fedkovych Chernivtsi National University

HIGH-PERFORMANCE DATA PROCESSING TECHNOLOGIES IN MODERN INFORMATION SYSTEMS BASED ON GPU

The rapid development of information systems and the exponential growth of data volumes in resource-intensive applications have created a critical need for high-performance data processing technologies. Modern information systems operating in big data environments require efficient The rapid development of information systems and the exponential growth of data volumes in resource-intensive applications have created a critical need for high-performance data processing technologies. Modern information systems operating in big data environments require efficient computational solutions to ensure high performance and scalability. Among contemporary approaches, the CUDA architecture developed by NVIDIA occupies a prominent place as a platform for implementing highly parallel computations on graphics processing units (GPUs). CUDA provides a programming model that enables the utilization of thousands of parallel threads, significantly increasing data processing performance compared to traditional CPU-oriented approaches. The memory hierarchy and thread management mechanisms allow optimization of data access, reduction of latency, and improvement of throughput, which are crucial for information systems dealing with large-scale datasets.

Sorting, as a fundamental data processing operation, is widely used in information systems, particularly in indexing, searching, and analytical data processing. Classical algorithms such as bubble sort, insertion sort, and quicksort are characterized by different computational complexities and performance levels. In particular, bubble sort has a complexity of $O(n^2)$ and is inefficient for large datasets, whereas algorithms with O(n log n) complexity are more suitable for practical applications. However, as data volumes increase, sequential implementations significantly decrease in performance. The use of GPUs helps overcome these limitations through massive parallelism, where data processing is performed simultaneously by many threads. To experimentally validate the effectiveness of the proposed approaches, a comparative analysis of sequential and parallel implementations of bitonic and bubble sort was conducted. Each experiment was performed ten times for different array sizes, and the average execution time was calculated. The results for bitonic sorting show a significant increase in efficiency of the GPU implementation with larger input data. For an array of size 2^{15}, execution time was 9.73 ms on the CPU and 3.82 ms on

the GPU. For size 2^{19}, these values were 412.75 ms and 14.04 ms, respectively, and for 2^{20}, 1012.52 ms and 21.75 ms. The largest tested array (2^{21}) showed times of 2262.45 ms on the CPU and 39.80 ms on the GPU. With increasing data size, speedup exceeds fifty times, indicating high scalability of GPU processing. Bubble sort shows a more complex performance pattern: for small arrays (2^{12}), the CPU may outperform due to parallelization overhead, but for 2^{15} speedup is ~5×, and for 2^{16}, more than twelve times.

Among algorithms designed for parallel processing, bitonic sort demonstrates high efficiency due to its regular computational structure and the possibility of independent element processing. This makes it well-suited for high-performance information technologies. Efficiency in parallel environments depends not only on asymptotic complexity but also on the degree of parallelization and memory access patterns. The obtained results indicate that the efficiency of data processing technologies depends on both hardware features and algorithm choice. Bitonic sort, due to its structure, is more suitable for GPU implementation, whereas bubble sort remains limited due to high computational complexity. In conclusion, using CUDA in information systems opens significant opportunities for improving large-scale data processing. Future research should focus on memory optimization, adaptive parallel algorithms, and integrating GPU computing into modern information systems, contributing to efficient solutions of complex applied problems.

References:

1. Ala'anzy M. A., Tolendi N., Baubek B., Algarni A. Performance evaluation of GPU-based parallel sorting algorithms // PLOS ONE. – 2026. – Vol. 21, No. 2. – e0342167. – DOI: 10.1371/journal.pone.0342167.

2. Yazici A., Gokahmetoglu H. Implementation of sorting algorithms with CUDA: An empirical study // International Journal of Applied Methods in Electronics and Computers. – 2016. – DOI: 10.18100/ijamec.53457.