УДК 004.8

А.В. Сачок
В.О. Денисюк

РЕАЛІЗАЦІЯ ПАРАЛЕЛЬНОГО АЛГОРИТМУ СОРТУВАННЯ БУЛЬБАШКОЮ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЇ CUDA

Вінницький національний технічний університет

Анотація

Розглянуто розробку паралельного алгоритму сортування бульбашкою з використанням технології CUDA та обчислювальних можливостей графічних процесорів. Проведено аналіз існуючих методів паралельного сортування та особливостей їх реалізації на архітектурі GPU з метою підвищення продуктивності обчислень. Обґрунтовано вибір технології CUDA як засобу реалізації паралельних обчислень, розглянуто модель програмування та особливості організації потоків і блоків. Розроблено програмну реалізацію паралельного алгоритму сортування бульбашкою та виконано тестування його швидкодії на різних наборах даних із порівнянням із послідовною реалізацією на CPU. Отримані результати демонструють доцільність використання графічних процесорів для прискорення обробки масивів даних та можуть бути використані для підвищення ефективності програм, що потребують інтенсивних обчислень.

Ключові слова: *паралельні обчислення, сортування бульбашкою, CUDA, GPU, продуктивність.*

Abstract

The development of a parallel bubble sort algorithm using CUDA technology and the computing capabilities of graphics processors is considered. An analysis of existing parallel sorting methods and the features of their implementation on GPU architecture is carried out with the aim of increasing computing performance. The choice of CUDA technology as a means of implementing parallel computing is justified, and the programming model and features of thread and block organization are considered. A software implementation of the parallel bubble sort algorithm has been developed, and its performance has been tested on various data sets with comparison to sequential implementation on the CPU. The results demonstrate the feasibility of using graphics processors to accelerate the processing of data arrays and can be used to improve the efficiency of programs that require intensive computing.

Keywords: *parallel computing, bubble sort, CUDA, GPU, performance.*

Вступ

Актуальність реалізації паралельного алгоритму сортування бульбашкою з використанням технології CUDA полягає у необхідності підвищення швидкодії обробки великих масивів даних у сучасних обчислювальних системах. Використання графічних процесорів дає змогу значно прискорити виконання обчислень за рахунок масового паралелізму, що є особливо важливим для задач аналізу даних, обробки сигналів, наукових обчислень та комп'ютерної графіки. Незважаючи на простоту алгоритму сортування бульбашкою, його паралельна реалізація дозволяє наочно продемонструвати ефективність використання архітектури GPU та технології CUDA.

Застосування паралельних обчислювальних архітектур дозволяє одночасно виконувати порівняння та перестановку елементів масиву у великій кількості потоків, що суттєво скорочує час виконання сортування. Технологія CUDA забезпечує ефективний розподіл обчислень між потоками та блоками, що сприяє масштабованості та підвищенню продуктивності алгоритму на графічному процесорі.

Метою роботи є розробка та дослідження паралельного алгоритму сортування бульбашкою з використанням технології CUDA з метою підвищення швидкодії та продуктивності обчислювальних систем.

Постановка задачі дослідження

Задачі дослідження полягають у вирішенні наступних питань:

- аналіз особливостей алгоритму сортування бульбашкою та можливостей його паралелізації;
- розробка паралельного алгоритму сортування бульбашкою з використанням технології CUDA;
- оптимальний розподіл обчислень між потоками та блоками графічного процесора;
- реалізація програмного модуля сортування на основі архітектури GPU;
- тестування розробленої програми та аналіз продуктивності отриманих результатів у порівнянні з послідовною реалізацією.

Виклад основного матеріалу

Сортування даних є однією з базових операцій в інформатиці та широко використовується в задачах аналізу даних, обробки сигналів, баз даних, комп'ютерної графіки та машинного навчання для попередньої підготовки і впорядкування інформації [1–3]. Алгоритм сортування бульбашкою належить до найпростіших методів сортування та має обчислювальну складність $O(n^2)$ що робить його малоефективним для обробки великих масивів даних у послідовному виконанні [4]. Проте завдяки можливості паралелізації операцій порівняння та обміну елементів, даний алгоритм може бути адаптований для виконання на паралельних обчислювальних архітектурах, зокрема на графічних процесорах [5].

Для обґрунтування доцільності використання технології CUDA в задачах паралельного сортування виконано дослідження архітектури графічних процесорів та принципів паралельних обчислень. CUDA (Compute Unified Device Architecture) є платформою та моделлю програмування, розробленою компанією NVIDIA, яка дозволяє використовувати GPU для виконання загальних обчислень з високим рівнем паралелізму [6, 7].

Графічний процесор складається з великої кількості обчислювальних ядер, здатних одночасно виконувати тисячі потоків. У моделі програмування CUDA обчислення організовуються у вигляді сітки (grid), що складається з блоків (blocks), які, у свою чергу, містять потоки (threads). Кожен потік виконує одну і ту ж програму (kernel), але працює з різними даними, що робить CUDA ефективним інструментом для масивно-паралельних задач, таких як сортування великих масивів [6, 7].

На відміну від послідовного виконання алгоритму сортування бульбашкою на центральному процесорі, паралельна реалізація на GPU дозволяє одночасно виконувати операції порівняння та перестановки елементів масиву у різних потоках. Це зменшує загальний час виконання програми, особливо при обробці великих обсягів даних. Проте ефективність такого підходу залежить від правильного розподілу потоків, використання пам'яті та синхронізації між потоками.

Основні переваги використання технології CUDA для паралельного сортування включають:

- масовий паралелізм, що дозволяє виконувати велику кількість операцій порівняння одночасно;
- ефективне використання апаратних ресурсів GPU;
- масштабованість, яка забезпечує підвищення продуктивності зі зростанням розміру вхідних даних;
- можливість оптимізації доступу до глобальної та спільної пам'яті.

У роботі розглянуто та використано кілька основних підходів до оптимізації алгоритму сортування бульбашкою з використанням CUDA.

- За рахунок паралелізації обчислень. Масив даних розподіляється між потоками GPU таким чином, що кожен потік виконує операції порівняння та обміну для окремих пар елементів. Це дозволяє значно скоротити кількість послідовних кроків алгоритму.
- За рахунок використання архітектури GPU. Організація потоків у блоки дозволяє ефективно використовувати апаратні можливості GPU. Синхронізація потоків у межах блоку забезпечує коректність виконання операцій сортування.

- За рахунок особливостей реалізації алгоритму. Алгоритм сортування бульбашкою було адаптовано для паралельного виконання шляхом поділу ітерацій на незалежні етапи, що зменшує кількість конфліктів при доступі до пам'яті та підвищує продуктивність.
- За рахунок оптимізації доступу до пам'яті. Використання спільної пам'яті GPU дозволяє зменшити затримки доступу до даних та підвищити швидкодію алгоритму. Обмін даними між потоками виконується після завершення відповідних етапів сортування з обов'язковою синхронізацією.

Програмно реалізовано задану задачу та описано всі компоненти коду [8].

Аналіз результатів тестування програми виконано для різної кількості елементів масиву та потоків (рис.1-3).

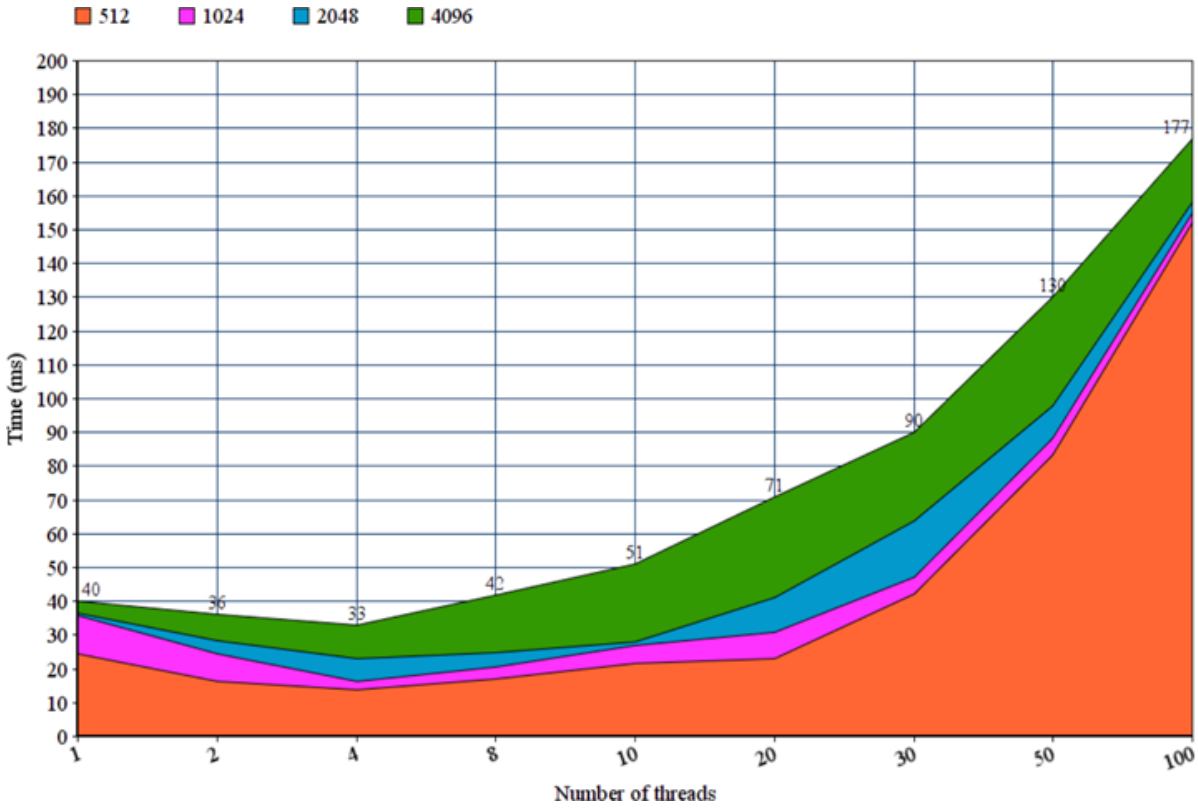

Рисунок 1 – Час виконання програми за різних вхідних даних

ArraySize	GPU Time (ms)	CPU Time (ms)
10	0.764	0.000800
50	0.307	0.022600
100	0.253	0.051400
5000	2.891	119.898600
10000	10.009	414.084900
100000	2137.717	N/A
500000	55682.563	N/A
1000000	246870.482	N/A

Рисунок 2 – Порівняння послідовної та паралельної реалізацій алгоритму

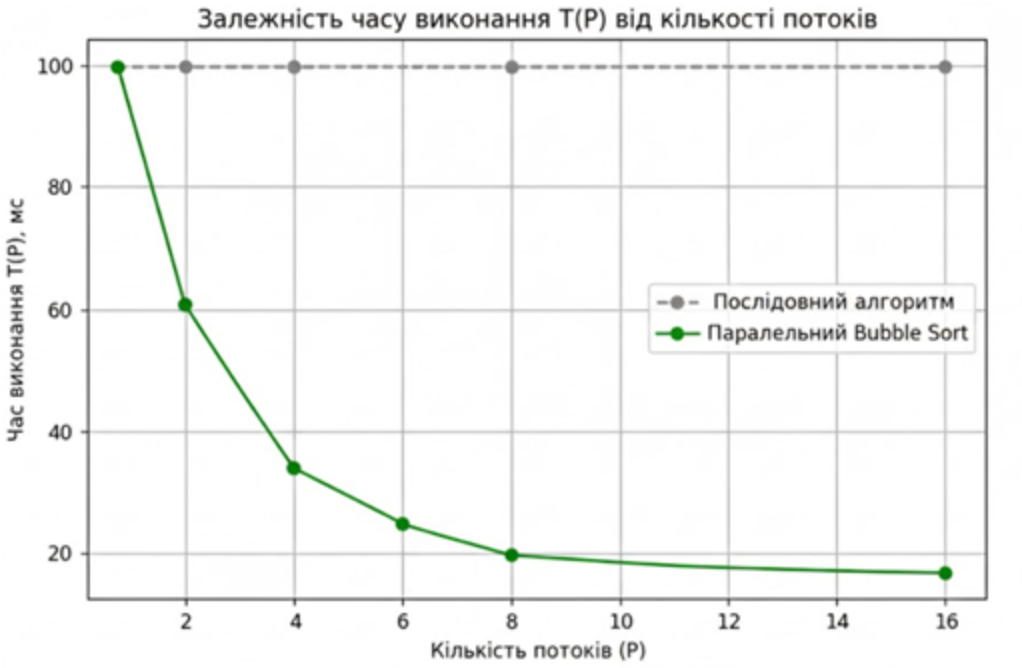

Рисунок 3 – Порівняння часу виконання алгоритму за різної кількості потоків

Проаналізувавши обчислені коефіцієнти прискорення та коефіцієнти ефективності для різних розмірів вхідних масивів, варіантів реалізації алгоритму (послідовний та паралельний) і різної кількості потоків GPU (рис. 1–3), можна зробити наступні висновки:

- при невеликих обсягах вхідних даних значення коефіцієнта прискорення та коефіцієнта ефективності є незначними, оскільки накладні витрати на організацію паралельних обчислень перевищують виграш від їх використання, тому збільшення кількості потоків практично не впливає на час виконання програми [9];
- збільшення кількості потоків при виконанні сортування для великих масивів даних призводить до зростання коефіцієнта прискорення, однак при цьому спостерігається зменшення коефіцієнта ефективності, що зумовлено витратами на синхронізацію потоків та обмін даними в пам’яті графічного процесора [10-11].

Отже, основними чинниками, які впливають на зміну таких показників, як коефіцієнт прискорення, коефіцієнт ефективності та загальний час виконання програми, є розмір вхідних даних (кількість елементів масиву), кількість задіяних потоків GPU, а також спосіб реалізації алгоритму сортування. При заміні базового алгоритму сортування бульбашкою на більш ефективні паралельні методи сортування якісні показники продуктивності можуть бути суттєво покращені.

Висновки

Досліджено алгоритм сортування бульбашкою як один із базових алгоритмів сортування, визначено його основні характеристики, переваги та недоліки, а також сфери застосування. Розглянуто можливості паралелізації даного алгоритму та особливості його реалізації на графічних процесорах. На основі аналізу літературних джерел досліджено архітектуру GPU та технологію CUDA як ефективний інструмент для організації паралельних обчислень, розглянуто модель програмування, структуру потоків і блоків та механізми доступу до пам’яті.

Програмно реалізовано паралельний алгоритм сортування бульбашкою з використанням CUDA та описано основні компоненти програмного коду. Проведено тестування розробленої програми на різних наборах вхідних даних, проаналізовано отримані результати та виконано порівняння з послідовною реалізацією алгоритму на центральному процесорі. Досліджено показники прискорення та ефективності паралельного виконання.

Визначено, що зі збільшенням кількості паралельних потоків на GPU спостерігається зростання

коефіцієнта прискорення до певного рівня, після чого ефективність зменшується через накладні витрати на синхронізацію та обмін даними між потоками. Отримані результати підтверджують доцільність використання графічних процесорів для прискорення обчислювально інтенсивних задач сортування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Parallel Sorting University of Porto, Faculty of Computer Science. URL: https://www.dcc.fc.up.pt/~ricroc/aulas/1516/cp/apontamentos/slides_sorting.pdf
2. Parallel Bubble Sort Using CUDA GeeksforGeeks. URL: https://www.geeksforgeeks.org/bubble-sort-using-cuda/
3. Singh R., Sharma P. Parallel Sorting Algorithms: Performance Evaluation and Comparative Analysis. Electronic edition. IEEE Xplore, 2022.
4. Rajasekaran S., Reif J. (Eds.). Handbook of Parallel Computing: Models, Algorithms and Applications. Boca Raton: CRC Press, 2010. 800 с.
5. Дорошенко А. Ю., Андон П. І., Яценко О. А., Жереб К. А. Алгебро-алгоритмічні моделі та методи паралельного програмування. Київ: Видавничий дім «Академперіодика», 2018. 192 с.
6. Narayanan S., Srinivasan K. Efficient DAG-based Task Scheduling in Parallel Computing. arXiv, 2020. 15 с. URL: https://arxiv.org/abs/2004.10908
7. Selim G. Akl. Parallel Sorting Algorithms. San Diego: Academic Press, 1993. 300 с.
8. PlantUML Editor. URL: https://plantuml-editor.kkeisuke.dev/
9. NVIDIA CUDA C Programming Guide. NVIDIA Corporation. URL: https://docs.nvidia.com/cuda/cuda-c-programming-guide/
10. Mahmood A., Al-Mamory S. Performance Evaluation of Parallel Bubble Sort on GPU Using CUDA. Journal of Computer Science, 2021.
11. Sanders J., Kandrot E. CUDA by Example: An Introduction to General-Purpose GPU Programming. Addison-Wesley, 2010. 310 с.

Сачок Андрій Володимирович – студент кафедри комп`ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: sachok95y@gmail.com;

Денисюк Валерій Олександрович – канд. техн. наук, доцент, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м.Вінниця, e-mail: vad64@i.ua.

Sachok Andrii Volodymyrovych – student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: sachok95y@gmail.com;

Denysiuk Valerii Olexandrovich – Ph.D., Assistant Professor, Assistant Professor of the Chair of Computer Science, Vinnytsia National Technical University, Vinnytsia, e-mail: vad64@i.ua