

ДОСЛІДЖЕННЯ ТА РОЗРОБКА ПАРАЛЕЛЬНОГО АЛГОРИТМУ СОРТУВАННЯ ЗА РОЗРЯДАМИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ C++ AMP

Вінницький національний технічний університет

Анотація

Досліджено, розроблено, оптимізовано та протестовано паралельний алгоритм сортування за розрядами з використанням C++ AMP. Оцінено його ефективності у порівнянні з послідовними підходами. Паралельна реалізація забезпечила прискорення виконання у кілька разів порівняно з послідовною; стабільність та коректність сортування; ефективне використання ресурсів CPU.

Ключові слова: Radix Sort, паралельний алгоритм, C++ AMP.

Abstract

A parallel algorithm for sorting by bits using C++ AMP was researched, developed, optimized, and tested. Its effectiveness was evaluated in comparison with sequential approaches. Parallel implementation provided several times faster execution compared to sequential; stability and correctness of sorting; efficient use of CPU resources.

Keywords: Radix Sort, parallel algorithm, C++ AMP.

Вступ

У сучасному світі інформаційних технологій обсяги даних, що підлягають обробці, зростають у геометричній прогресії. Це вимагає ефективних та високошвидкісних алгоритмів обробки, зокрема сортування. Одним з найефективніших алгоритмів для числових масивів є сортування за розрядами (Radix Sort), особливо у випадках роботи з великими наборами даних [1].

Використання паралельних обчислень дозволяє значно скоротити час обробки, розподіляючи навантаження між ядрами CPU або потоками на GPU. Вибір технології C++ AMP для реалізації Radix Sort обумовлений орієнтацією цієї технології на масивно-паралельну обробку, яка доступна на GPU.

Постановка задачі дослідження

Метою дослідження є розробка, оптимізація та тестування паралельного алгоритму сортування за розрядами з використанням C++ AMP, а також оцінка його ефективності у порівнянні з послідовними підходами.

Основні завдання дослідження такі:

- аналіз існуючих реалізацій паралельного сортування;
- розробка математичної моделі алгоритму;
- створення потокового графа виконання;
- вибір відповідної технології реалізації;
- програмування та оптимізація алгоритму;
- проведення тестування для аналізу продуктивності.

Виклад основного матеріалу

Алгоритм Radix Sort — це стабільний, непорівняльний алгоритм сортування, що обробляє елементи за окремими розрядами чисел, починаючи з найменш значущого (LSD) до найбільш значущого (MSD). Складність алгоритму в класичному випадку — $O(n * k)$, де n — кількість елементів, k — кількість розрядів. У паралельному варіанті обчислення частот, сортування за

кожним розрядом та злиття частково відсортованих блоків виконуються одночасно потоками. Це дозволяє значно зменшити час виконання (крокова складність — $O(\log n)$, робоча — $O(n)$) [2].

Первинно було заплановано використання C++ AMP як платформи для паралельного програмування на GPU [3]. Проте, через її обмежену підтримку на сучасних системах, реалізація була здійснена за допомогою стандартних потоків C++ (`std::thread`), що надали більшу гнучкість та портативність. У реалізації застосовувалися також інструменти [4]: Visual Studio як середовище розробки; PlantUML — для побудови UML-діаграм; `mutex` та `lock_guard` — для синхронізації потоків.

Програмна реалізація включає кілька класів:

- RadixSort — керує процесом сортування та потоками;
- DataBlock — представляє окремий блок даних;
- FrequencyCounter — обчислює частоти значень;
- CountingSort — реалізує сортування одного блоку;
- Synchronizer — відповідає за міжпотоківу взаємодію.

Підхід базується на розподілі даних на блоки, сортуванні кожного в окремому потоці та подальшому злитті результатів з використанням синхронізації [5].

Результати дослідження.

Тестування довели роботу спроможність обраних методів та результативність розробки.

Для тестування були сформовані різні тестові набори: малі, середні, великі масиви; набори з однаковими елементами, великі числа тощо. Всі тести пройдено успішно. Паралельна реалізація забезпечила:

- прискорення виконання у кілька разів порівняно з послідовною;
- стабільність та коректність сортування;
- ефективне використання ресурсів CPU.

Висновки

Успішно реалізовано ефективний паралельний алгоритм сортування за розрядами. Програмна реалізація підтвердила переваги паралельного підходу при роботі з великими обсягами даних.

Отримані результати можуть бути використані у програмних продуктах, які потребують високошвидкісного сортування: обробка баз даних, системи обробки зображень, пошукові системи.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Сортування за розрядами. URL:https://uk.wikipedia.org/wiki/Сортування_за_розрядами .
2. Паралельні обчислення. URL:https://uk.wikipedia.org/wiki/Паралельні_обчислення .
3. C++ AMP Overview. URL:<https://learn.microsoft.com/en-us/cpp/parallel/amp/cpp-amp-overview?view=msvc170> .
4. C++ documentation. URL:<https://learn.microsoft.com/en-us/cpp/cpp/?view=msvc-170> .
5. Алгоритм сортування за розрядами | Radix | Algorithms and Data structures. URL:<https://www.youtube.com/watch?v=lCgqzCB2Zo> .

Северин Ілля Володимирович — студент кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: severinilla2019@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.

Severyn Ilya Vladimirovich — student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: severinilla2019@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