УДК 004.8

В.О.Морозов
В.О. Денисюк

ГІБРИДНА РЕАЛІЗАЦІЯ АЛГОРИТМУ СОРТУВАННЯ ЗА РОЗРЯДАМИ З ВИКОРИСТАННЯМ DIRECTCOMPUTE

Вінницький національний технічний університет

Анотація

У роботі розглянуто розробку гібридного алгоритму сортування за розрядами (Radix Sort), у якому обчислення часткових операцій виконуються на графічному процесорі (GPU) засобами DirectCompute, а завершальний стабільний етап формування відсортованого масиву виконується центральним процесором (CPU). Проаналізовано існуючі сортувальні алгоритми, визначено їх переваги й недоліки у контексті паралельних обчислень, обґрунтовано вибір Radix Sort як базового методу через його природну декомпозицію на незалежні підзадачі. Розроблено структуру програмного модуля, створено класи і програмну логіку взаємодії CPU–GPU, побудовано програмну реалізацію на основі DirectCompute та проведено тестування продуктивності на різних обсягах вхідних даних. Показано, що гібридний підхід забезпечує коректність сортування й здатний покращувати продуктивність для великих масивів, водночас демонструючи характерні ефекти амортизації накладних витрат при зростанні розміру масиву.

Ключові слова: *сортування за розрядами, DirectCompute, GPU, паралельний алгоритм, гібридне сортування.*

Abstract

The paper addresses the development of a hybrid radix sort algorithm in which partial operations are performed on a graphics processing unit (GPU) using DirectCompute, while the final stable redistribution is carried out on the central processing unit (CPU). Existing sorting methods are analyzed, their applicability to parallel processing is discussed, and Radix Sort is justified as a suitable algorithm due to its natural decomposition properties. A software module structure was developed, CPU–GPU interaction was implemented using DirectCompute, and performance testing was conducted on input arrays of various sizes. The results demonstrate that the hybrid approach provides correct sorting and may improve performance on large datasets, while also showing amortization effects of GPU-related overhead as data size increases.

Keywords: *radix sort, DirectCompute, GPU computing, hybrid algorithm, parallel sorting.*

Вступ

Проблема ефективної обробки великих масивів даних залишається однією з ключових у сучасних інформаційних системах. Сортування є базовою операцією багатьох алгоритмів, а тому вибір оптимального методу для великих наборів даних має суттєве практичне значення.

Алгоритм Radix Sort вирізняється тим, що його трудомісткість визначається кількістю розрядів чисел і не залежить від порівнянь, що робить його перспективним для паралелізації. Використання DirectCompute дозволяє перекласти обчислення гістограми цифр на GPU, який краще пристосований до однотипних операцій над великими наборами даних.

Актуальність даного дослідження полягає у створенні гібридної CPU–GPU моделі сортування, яка поєднує паралелізм GPU і стабільність традиційної реалізації Radix Sort на CPU.

Постановка задачі дослідження

Основні задачі роботи полягали у розв’язанні таких питань:

- аналіз існуючих методів сортування та визначення їх придатності для паралельної реалізації;
- обґрунтування вибору DirectCompute як технології для прискорення обчислень на GPU;
- розробка програмної архітектури гібридного Radix Sort, що включає розподіл функцій

між CPU та GPU;

- створення і реалізація обчислювального шейдера для формування гістограми цифр;
- проведення тестування продуктивності на масивах різного розміру;
- аналіз отриманих результатів, порівняння поведінки алгоритму на малих і великих наборах даних.

Виклад основного матеріалу

Алгоритми сортування є одними з ключових процедур у комп'ютерній науці, оскільки вони широко застосовуються у базах даних, інформаційних системах, комп'ютерній графіці, мережевих застосунках та системах обробки великих масивів даних [1–4]. Серед класичних методів сортування виділяють алгоритми, що базуються на порівняннях (QuickSort, MergeSort, HeapSort) зі складністю порядку O(n log n), а також лінійні алгоритми, такі як сортування за розрядами (Radix Sort), для якого часова складність визначається як O(k·n) і є особливо ефективною для упорядкування числових масивів із обмеженим діапазоном значень [1, 3, 5].

На основі аналізу літературних джерел було досліджено особливості алгоритму Radix Sort, зокрема принцип поцифрового опрацювання чисел, стабільність сортування та залежність продуктивності від кількості розрядів і розміру вхідного масиву [1, 5]. Окрему увагу приділено питанням побудови паралельних алгоритмів та оцінці їх складності, а також можливостям використання сучасних паралельних платформ для прискорення обробки даних [6–9].

У роботі розглянуто гібридний підхід до реалізації сортування за розрядами, у якому поєднується використання графічного процесора (GPU) та центрального процесора (CPU). Обчислення гістограми цифр для поточного розряду виконується на графічному процесорі за допомогою технології DirectCompute, що дозволяє задіяти масовий паралелізм, характерний для GPU. Подальший стабільний перерозподіл елементів здійснюється на центральному процесорі, що спрощує реалізацію і забезпечує коректність підсумкового впорядкування.

Технологія DirectCompute, інтегрована до складу DirectX 11, надає можливість виконання загальновживаних обчислень на графічному процесорі за допомогою обчислювальних шейдерів. У програмній реалізації використовується обчислювальний шейдер HistogramCS, написаний мовою HLSL, який виконує поелементне опрацювання масиву та атомарне оновлення глобальної гістограми цифр. Для організації доступу до даних застосовуються структуровані буфери (StructuredBuffer, RWStructuredBuffer), константний буфер із параметрами N та Exp, а також відповідні SRV/UAV-представлення ресурсів [7–9].

Сторону CPU реалізовано мовою C++ із використанням стандартної бібліотеки та засобів роботи з пам'яттю й контейнерами [8, 10]. Центральний процесор ініціалізує пристрій Direct3D, завантажує обчислювальний шейдер, формує вхідні дані, передає масив на GPU, а після виконання шейдера зчитує побудовану гістограму та виконує етап стабільного перерозподілу елементів згідно з обчисленою кумулятивною сумою.

Для оцінки продуктивності гібридного алгоритму було проведено серію експериментів із масивами випадкових чисел різного розміру (10 000, 20 000, 40 000 та 80 000 елементів). У кожному випадку фіксувався повний час виконання циклу Radix Sort, що охоплює всі розряди чисел, а також перевірялася коректність результату за допомогою стандартної функції std::is_sorted [8].

Узагальнені результати вимірювань подано у таблиці 1.

Як показує аналіз результатів, збільшення розміру масиву не призводить до пропорційного зростання часу обробки; навпаки, при переході до більших масивів спостерігається відносне зменшення часу на один елемент. Така поведінка пояснюється тим, що при малих N помітну частку загального часу займають накладні витрати на ініціалізацію пристрою, передачу даних і синхронізацію між CPU та GPU. У випадку великих масивів ці накладні витрати амортизуються, а графічний процесор завантажується більш повно, що підвищує ефективність використання апаратних ресурсів [6–9].

Таблиця 1 – Час роботи паралельного та послідовного сортувань при різних обсягах даних

Розмір масиву	Час сортування послідовно, ms	Час сортування паралельно, ms	Коректність
10 000	1.8248	5.4256	Відсортовано правильно
100 000	17.9046	19.0129	Відсортовано правильно
1 000 000	197.4250	169.7650	Відсортовано правильно

Отримані результати підтверджують доцільність застосування гібридної реалізації Radix Sort із використанням DirectCompute для сортування великих числових масивів, де переваги масового паралелізму GPU компенсують накладні витрати, пов'язані з організацією взаємодії між процесорами.

Висновки

Досліджено алгоритм сортування за розрядами як один із ефективних методів лінійного сортування числових даних, розглянуто його місце серед інших алгоритмів сортування, зокрема методів на основі порівнянь, та проаналізовано теоретичні оцінки складності й області застосування [1–5]. На основі літературних джерел опрацьовано питання побудови паралельних та гібридних алгоритмів, а також використання сучасних апаратних засобів для прискорення обробки даних [6– 9].

У роботі розроблено та програмно реалізовано гібридний алгоритм сортування за розрядами, у якому побудова гістограми цифр для поточного розряду виконується на графічному процесорі за допомогою DirectCompute, а стабільний перерозподіл елементів – на центральному процесорі. Описано структуру програмного модуля, взаємодію між CPU та GPU, а також основні компоненти коду мовами C++ та HLSL [7-10].

Проведено тестування роботи алгоритму на масивах різного розміру (10 000–80 000 елементів). Результати експериментів показали, що гібридна реалізація забезпечує коректне сортування вхідних даних, а час виконання залишається прийнятним навіть для десятків тисяч елементів. Виявлено, що зі збільшенням обсягу даних накладні витрати на ініціалізацію та синхронізацію відносно зменшуються, що позитивно впливає на ефективність використання графічного процесора [6–9].

Отже, реалізований гібридний підхід до Radix Sort із використанням DirectCompute може бути рекомендований для застосування в програмних системах, де необхідне швидке сортування великих масивів числових даних, а також може слугувати основою для подальшого розширення й інтеграції з іншими паралельними алгоритмами обробки інформації.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Про алгоритми сортування. URL: https :// foxminded . ua / alhorytmy - sortuvannia /
2. Selection sort. URL: https://en.wikipedia.org/wiki/Selection_sort
3. Quicksort: історія виникнення та розвитку "найшвидшого" алгоритму сортування. URL: https://phm.cuspu.edu.ua/nauka/naukovopopuliarni-publikatsii/824-quicksort-istoriia-vynyknennia-ta-rozvytkunaishvydshoho-alhorytmu-sortuvannia.html
4. Сортування злиттям: алгоритм, переваги і особливості. URL: https :// kafedra . com . ua / sortuvannya - zlyttyam - algorytm - perevagy - i - osoblyvosti /
5. Radix Sort. URL: https :// www . ritambhara . in / radix - sort /
6. Паралельні алгоритми та їх складність. URL: https://javarush.com/ua/quests/lectures/ua.javarush.python.core.lecture.level20.lecture05
7. Жуков І.А. Паралельні та розподілені обчислення. Лабораторний практикум / І.А. Жуков, О.В. Корочкін. К. : Корнейчук, 2008. 224 с.
8. Проектування та аналіз обчислювальних алгоритмів: Вступ до алгоритмів [Електронний ресурс]: навчальний посібник для студ. спеціальності 122 «Комп'ютерні науки» / І. В. Федорін; КПІ ім. Ігоря Сікорського. Електронні текстові дані (1 файл: 1,97 Мбайт). Київ: КПІ ім. Ігоря Сікорського, 2022. 115 с.
9. Методичні вказівки до лабораторних робіт з навчальної дисципліни «Паралельні та розподілені обчислення» (частина 1) для здобувачів вищої освіти першого (бакалаврського) рівня за освітньопрофесійною програмою «Комп'ютерна інженерія» спеціальності 123 «Комп'ютерна інженерія» денної і заочної форм навчання [Електронне видання] / Бойчура М. В., Шатний С. В., Шатна А. В. Рівне : НУВГП, 2023. 49 с.
10. C++. Довідкова документація. URL: https :// w 3 schoolsua . github . io / cpp / cpp ref reference . html # gsc . tab =0

Морозов Владислав Олегович – студент кафедри комп`ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: vipmiallo@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.

Morozov Vladislav Olegovych – student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: vipmiallo@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