

РЕАЛІЗАЦІЯ ПАРАЛЕЛЬНОГО АЛГОРИТМУ СОРТУВАННЯ QUICK SORTING НА DIRECT COMPUTE

Вінницький національний технічний університет

Анотація

Розглянуто розробку та програмну реалізацію оптимізованого технологією Direct Compute паралельного алгоритму сортування класу Parallel QuickSort із використанням технологій платформи .NET та мови програмування C#. Проаналізовано роль паралельного сортування в сучасних комп'ютерних системах, виконано математичне моделювання та структурний аналіз алгоритму, синтезовано його потоковий граф. Обґрунтовано вибір програмних засобів реалізації, розроблено UML-діаграму класів і блок-схему алгоритму. Створено програмну реалізацію оптимізованого паралельного алгоритму та проведено тестування його продуктивності для масивів різного розміру. Використання отриманих результатів дозволяє підвищити швидкодію, продуктивність та ефективність сортування великих обсягів даних у багатопотокових обчислювальних середовищах.

Ключові слова: паралельний алгоритм, паралельне сортування, ParallelQuickSort, Direct Compute, потоковий граф, C#.

Abstract

The text describes the development and software implementation of an optimized DirectCompute-based parallel sorting algorithm of the Parallel QuickSort class using the .NET platform and C#. It analyzes the role of parallel sorting in modern computer systems, performs mathematical modeling and structural analysis of the algorithm, and synthesizes its data-flow graph. The choice of implementation tools is justified, and a UML class diagram and algorithm flowchart are developed. A software implementation of the optimized parallel algorithm is created, and its performance is tested on arrays of various sizes. The obtained results can improve speed, performance, and efficiency when sorting large data volumes in multithreaded computing environments.

Keywords: parallel algorithm, parallel sorting, ParallelQuickSort, DirectCompute, data-flow graph, C#.

Вступ

Актуальність паралельного сортування структур даних зумовлена стрімким зростанням обсягів інформації та широким використанням графічних прискорювачів (GPU) для обчислень загального призначення. У сучасних програмних комплексах, зокрема в системах BigData, базах даних, фінансових застосунках та обчислювальних платформах, послідовні алгоритми сортування часто не забезпечують необхідного рівня продуктивності. Використання паралельних алгоритмів на базі технології DirectCompute дозволяє ефективніше задіяти обчислювальні ресурси GPU та значно скоротити час виконання операцій над великими масивами даних [1, 2].

Метою роботи є розробка, структурний аналіз та програмна реалізація оптимізованого паралельного алгоритму сортування на основі бітонічного сортування із використанням технології DirectCompute, платформи .NET та мови програмування C# [3].

Постановка задачі дослідження

Для досягнення поставленої мети у роботі необхідно виконати аналіз предметної області паралельного сортування на GPU, здійснити математичне моделювання бітонічного алгоритму сортування для архітектури DirectCompute, синтезувати його потоковий граф, спроектувати програмну реалізацію з використанням UML-діаграм, реалізувати оптимізований паралельний алгоритм із використанням compute shader та провести тестування з порівняльним аналізом отриманих результатів відносно послідовної реалізації.

Виклад основного матеріалу

Паралельні алгоритми сортування на GPU є важливою складовою сучасних обчислювальних систем, оскільки дозволяють значно зменшити час обробки великих масивів даних за рахунок масового паралелізму графічних прискорювачів. У роботі розглянуто бітонічний алгоритм сортування, адаптований для виконання на GPU через технологію DirectCompute, який характеризується регулярною структурою обчислень та ефективним використанням паралельних потоків обробки.

На основі математичного моделювання досліджено часову складність алгоритму, сформульовано рекурентні співвідношення часу виконання на GPU архітектурі, визначено критичний шлях, висоту та ширину потокового графу, а також потенційний ступінь паралельності. Показано, що бітонічне сортування характеризується складністю $O(\log^2 n)$ порівнянь із можливістю виконання на масивно-паралельній архітектурі GPU, що забезпечує високий рівень ефективності для великих обсягів даних.

У рамках структурного аналізу виконано синтез потокового графу бітонічного алгоритму сортування на DirectCompute із використанням матриці суміжності та інших матричних подань, що дозволило формально оцінити можливості оптимізації GPU обчислень та мінімізації передачі даних між CPU та GPU. Для програмної реалізації розроблено UML-діаграму класів та блок-схему алгоритму, які відображають логіку ініціалізації Direct3D11 пристрою, компіляції HLSL compute shader, управління GPU буферами, виконання паралельних етапів сортування та синхронізації результатів.

Алгоритм реалізовано мовою C# з використанням бібліотеки SharpDX для роботи з DirectCompute API. Compute shader написано мовою HLSL (High-Level Shading Language) та налаштовано на виконання 256 паралельних потоків у робочій групі. З метою підвищення ефективності застосовано оптимізацію роботи з GPU пам'яттю, використання structured buffers для прямого доступу до даних, а також мінімізацію передачі даних між CPU та GPU через використання staging buffers для зчитування результатів.

Висновки

Проведене тестування програмної реалізації підтвердило коректність роботи оптимізованого паралельного алгоритму сортування на DirectCompute та його придатність до використання в GPU-прискорених обчислювальних середовищах. Встановлено, що при обробці великих масивів даних GPU-реалізація забезпечує значне скорочення часу виконання у порівнянні з послідовним CPU варіантом, демонструючи прискорення у декілька разів завдяки масовому паралелізму графічного прискорювача.

Отже, у роботі розв'язано всі поставлені завдання: виконано аналіз предметної області паралельного сортування на GPU, здійснено математичне моделювання та структурний аналіз бітонічного алгоритму сортування для DirectCompute, синтезовано його потоковий граф, побудовано UML-діаграми та блок-схеми архітектури системи, а також розроблено й досліджено програмну реалізацію оптимізованого паралельного алгоритму сортування з використанням технології DirectCompute та порівняльним аналізом ефективності відносно CPU-реалізації.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Cederman D., Tsigas P.. GPU-Quicksort: A Practical Quicksort Algorithm for Graphics Processors. URL: <https://scispace.com/pdf/gpu-quicksort-a-practical-quicksort-algorithm-for-graphics-11055ffzti.pdf>
2. Sintorn E., Assarsson U.. Fast Parallel GPU-Sorting Using a Hybrid Algorithm. URL: <https://www.cse.chalmers.se/~uffe/hybridsortElsevier.pdf>
3. Stephen Cleary. Concurrency in C# Cookbook: Asynchronous, Parallel, and Multithreaded Programming. 2nd Edition. URL: <https://dokumen.pub/concurrency-in-c-cookbook-asynchronous-parallel-and-multithreaded-programming-2nbsped-149205450x-9781492054504.html>

Доценко Максим Денисович – студент кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: maksimilianeux@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.

Dotsenko Maksym Denysovych – student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: maksimilianeux@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