

ПАРАЛЕЛЬНА РЕАЛІЗАЦІЯ МАТРИЧНОГО МНОЖЕННЯ ЗА ДОПОМОГОЮ BLOCK MATRIX З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ CUDA

Вінницький національний технічний університет

Анотація

У дослідженні розглянуто розробку паралельного алгоритму матричного множення за допомогою Block Matrix. Розглянуто питання аналізу матричного множення за допомогою Block Matrix, обґрунтовано засоби розробки програмного модуля, розроблено діаграми класів програмного модуля, обґрунтовано вибір програмного засобу реалізації, створено програмну реалізацію та проведено тестування програмної реалізації модуля паралельного алгоритму матричного множення за допомогою Block Matrix. Реалізація програмного модуля дозволяє побачити різницю у часі обчислення великих матриць на центральному процесорі (CPU) та графічному процесорі (GPU) звичайним методом та методом Штрассена за допомогою паралельного виконання, використовуючи технологію CUDA. В процесі аналізу результатів визначено прискорення GPU над CPU, оптимальний розмір блоку для GPU обчислень, ефективність методу Штрассена та сформовано рекомендацію щодо оптимального методу для різних розмірів матриць. Одержані результати можливо використовувати у різноманітних алгоритмах та програмних засобах для розуміння роботи матричного множення за допомогою Block Matrix та збільшення їх швидкодії.

Ключові слова: паралельний алгоритм, матричне множення, Block Matrix, технологія CUDA.

Abstract

The study analyzes the development of a parallel matrix multiplication algorithm using Block Matrix. It substantiates the development of a software module, presents class diagrams, explains the choice of software implementation, and describes the implementation and testing of the parallel matrix multiplication module using Block Matrix. The software module enables the comparison of computation times for large matrices on a central processing unit (CPU) and a graphics processing unit (GPU) using both the conventional and Strassen methods with parallel execution via CUDA technology. The study determined the GPU acceleration over the CPU, the optimal block size for GPU calculations, and the efficiency of the Strassen method, along with a recommendation for the optimal method depending on matrix size. The obtained results can be applied in various algorithms and software tools to enhance the understanding of matrix multiplication with Block Matrix and improve computational efficiency.

Keywords: parallel algorithm, matrix multiplication, Block Matrix, CUDA technology.

Вступ

Числа в матрицях можуть представляти дані, а також вони можуть представляти математичні рівняння. У багатьох інженерних програмах, чутливих до часу, множення матриць може дати швидкі, але хороші наближення набагато складніших обчислень [1]. Блочне множення матриць (Block Matrix) – це потужний метод, який дозволяє ефективно обчислювати добутки великих матриць, розділяючи їх на менші підматриці (блоки). Цей метод забезпечує більшу ефективність в обчисленнях завдяки можливості паралельної обробки блоків, що особливо корисно при роботі з великими масивами даних на графічних процесорах [2, 3].

В роботі досліджено матричне множення за допомогою блочних матриць двома методами на GPU й порівняне з виконанням цих ж обчислень на CPU, надаючи корисну інформацію та рекомендації про використання пристроїв та методів обчислення, що раніше не було наявне в вільному доступі.

Постановка задачі дослідження

При розв'язанні задачі паралельної реалізації матричного множення за допомогою Block Matrix необхідно:

- провести математичне моделювання матричного множення за допомогою Block Matrix за стандартним методом та методом Штрассена;
- розробити паралельну реалізацію алгоритму матричного множення за допомогою Block Matrix;
- реалізувати програмну реалізацію оптимізованого паралельного алгоритму матричного множення за допомогою Block Matrix та виконати детальний аналіз результатів її тестування.

Виклад основного матеріалу

Дослідження зосереджене на визначенні прискорення GPU над CPU при використанні матричного множення за допомогою Block Matrix, оптимального розміру блоку для GPU обчислень, ефективності методу Штрассена та сформуванні рекомендації щодо оптимального методу для різних розмірів матриць

Проведено аналіз предметної області матричного множення за допомогою Block Matrix. Підтверджено важливість матричного множення: центральна операція в усіх обчислювальних програмах лінійної алгебри, вирішення лінійних рівнянь, цифрове зображення та декодування відео, обробка цифрового звуку, представлення графів у матричній формі [1]. Виявлено перевагу стандартного методу над іншими для реалізації паралельних обчислень та визначено метод Штрассена як перспективний через його меншу обчислювальну складність [4, 5].

Для стандартного методу матричного множення за допомогою Block Matrix визначено, що кожен потік може бути призначений для обробки одного блоку, що забезпечує виконання множення матриць у паралельному режимі. Завдяки цьому час виконання зростає лінійно по відношенню до кількості доступних потоків, що робить GPU ідеальним вибором для таких обчислень.

В процесі розгляду способів покращення паралельного алгоритму розглянуто підвищення його арифметичної ефективності за допомогою методу Штрассена, що потребує 0,6 арифметичних затрат від стандартного алгоритму [4, 5].

Реалізовано програму оптимізованого потокового графу паралельного алгоритму засобами мови програмування C++ з використанням технології CUDA. Досліджено ефективність матричного множення за допомогою Block Matrix з використанням GPU та CPU з різними розмірами матриць та зміною основних параметрів. Експериментальні результати демонструють суттєве прискорення виконання порівняно з послідовною версією алгоритму, особливо на великих матрицях. Аналіз отриманих результатів показує, що GPU значно ефективніший за CPU для матричного множення, показуючи прискорення в сотні разів для великих матриць. Розуміння ефективності паралельного матричного множення на GPU відкриває перспективи для подальшого вдосконалення алгоритму під графічні процесори та його застосування в розподілених системах.

Результати дослідження.

Проведене тестування паралельної реалізації алгоритму матричного множення за допомогою Block Matrix показало значне підвищення швидкості порівняно з послідовною версією (рис. 1). При тестуванні відбувались зміни розмірів матриць від 64 на 64 до 4096 на 4096 при розміру блоку потоків CUDA від 8 до 32 та порогу використання алгоритму Штрассена від 64 до 512.

На основі всіх результатів можливо зробити такі узагальнення.

1. GPU значно ефективніший за CPU для матричного множення при розмірах матриць більше 128x128, показуючи прискорення в сотні разів для великих матриць.
2. Оптимальний розмір блоку для GPU обчислень становить 16, що забезпечує найкращий баланс між використанням кеш-пам'яті та паралельною обробкою даних.
3. Метод Штрассена показує різну ефективність на CPU та GPU. На CPU він стає ефективнішим за стандартний метод при великих розмірах матриць (від 2048x2048). На GPU він

залишається менш ефективним через додаткові накладні витрати на операції з пам'яттю та синхронізацію.

4. Вибір оптимального методу залежить від розміру вхідних матриць, доступного обладнання (CPU/GPU), вимог до швидкодії.

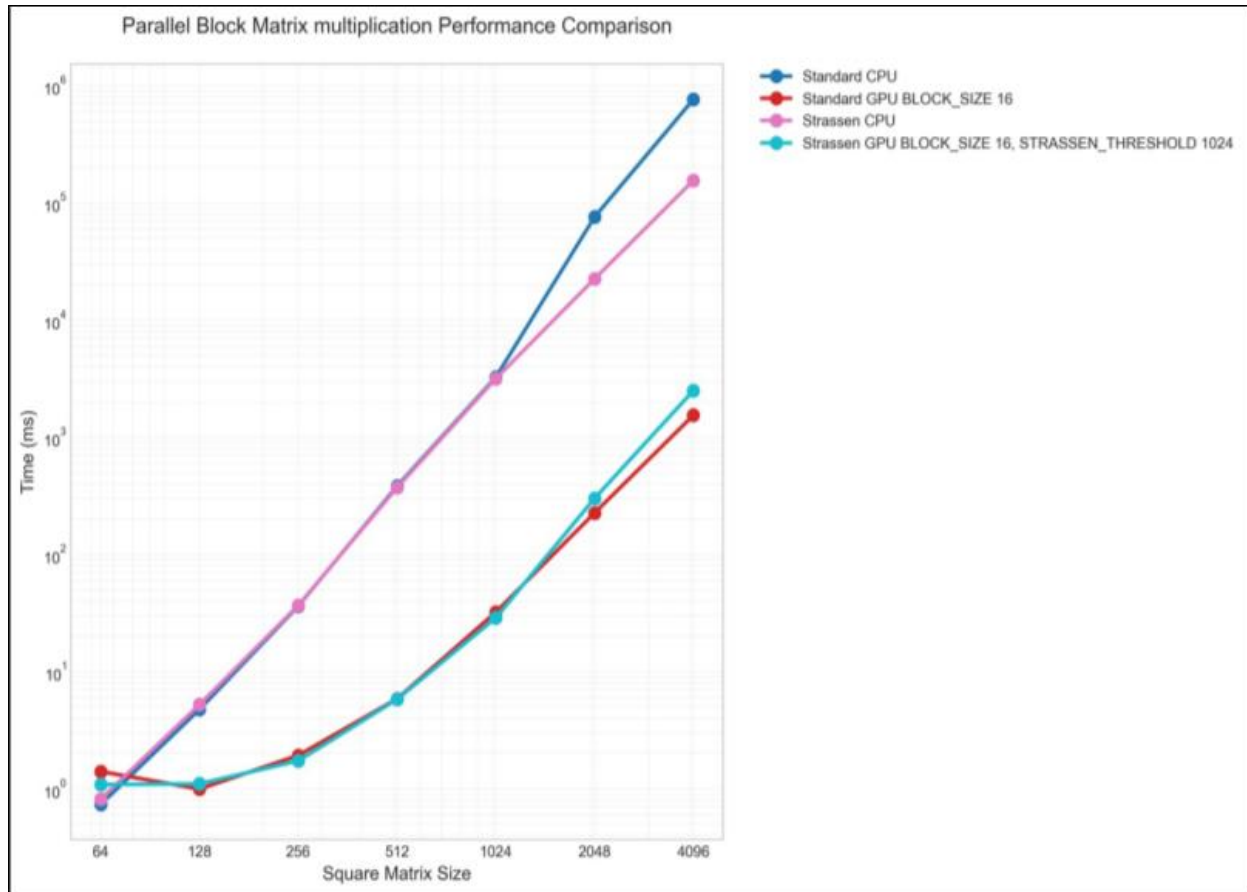


Рисунок 1 – Порівняння часу множення матриць на CPU та GPU

Для практичного використання сформульовані рекомендації.

1. Для малих матриць (до 128×128) – використовувати CPU зі стандартним методом.
2. Для середніх та великих матриць – використовувати GPU зі стандартним методом.
3. При відсутності GPU для дуже великих матриць – розглянути використання методу Штрассена на CPU.

Висновки

Дослідження підтверджує значне прискорення GPU над CPU при виконанні матричного множення за допомогою Block Matrix. Визначено оптимальний розмір блоку для GPU обчислень, ефективність методу Штрассена та сформовано рекомендації щодо оптимального методу для різних розмірів матриць.

Одержані результати можливо використовувати у різноманітних алгоритмах та програмних засобах для розуміння роботи матричного множення за допомогою Block Matrix та збільшення їх швидкодії. Подальші дослідження можуть бути зосереджені на ще більшому вдосконаленні алгоритму під графічні процесори та його застосування в розподілених системах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Matrix Multiplication-Why is it a big deal? Medium. URL: <https://charchithowitzer.medium.com/matrix-multiplication-why-is-it-a-big-deal-cc8ef7490008>.
2. Block Matrix Multiplication. The Ohio State University – Ximera team. URL: <https://ximera.osu.edu/la/LinearAlgebra/MAT-M-0023/main>.
3. Matrix multiplication. Wikipedia. URL: https://en.wikipedia.org/wiki/Matrix_multiplication.
4. Complexity of matrix multiplication. Wikipedia. URL: https://en.wikipedia.org/wiki/Computational_complexity_of_matrix_multiplication.
5. Операції над матрицями і прямі методи. URL: http://om.univ.kiev.ua/users_upload/15/upload/file/cm_lecture_07.pdf.

Иценко Гліб Андрійович – студент кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: glebishshenko1000@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.

Ishchenko Hlib Andriyovych – student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: glebishshenko1000@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