

ЗАСТОСУВАННЯ БАГАТОРІВНЕВОГО ПАРАЛЕЛІЗМУ ДЛЯ ПАРАЛЕЛЬНО-ІЄРАРХІЧНОГО ОБРОБЛЕННЯ ПЛЯМОПОДІБНИХ ЗОБРАЖЕНЬ

Вінницький національний технічний університет

Анотація

Удосконалено модель організації обчислювального процесу паралельно-ієрархічного перетворення шляхом застосування концепції багаторівневого паралелізму, що забезпечило підвищення швидкодії оброблення плямоподібних зображень

Ключові слова: паралельні обчислення, GPGPU, паралельно-ієрархічне перетворення, багаторівневий паралелізм.

Abstract

Organization model of calculation process of parallel-hierarchical transformation was improved by using concept of multi-level parallelism, what has provided increase of processing performance for spotlike images.

Keywords: parallel computing, GPGPU, parallel-hierarchical transformation, multi-level parallelism.

Вступ

Наразі існує велика кількість технологій паралельних обчислень, жодна з яких не є універсальною та має свої переваги, недоліки та області застосування [1]. Це накладає певні обмеження при використанні паралельних технологій для реалізації комплексних програмних систем, які повинні вирішувати широке коло задач, або працювати з великим різноманіттям даних. Одним з можливих та найбільш перспективних рішень є концепція багаторівневого паралелізму, яка передбачає комплексне застосування існуючих технологій на основі гетерогенного програмно-апаратного забезпечення з метою найбільш повного використання переваг паралельних обчислень.

Метою роботи є удосконалення моделі організації обчислювального процесу паралельно-ієрархічного перетворення шляхом застосування концепції багаторівневого паралелізму для підвищення швидкодії оброблення плямоподібних зображень.

Результати дослідження

До комплексних програмних систем можливо віднести й спеціалізовані системи для обробки зображень, які, як правило, повинні бути в змозі працювати з наборами зображень різної розмірності в реальному часі або наближеному до нього, що висуває високі вимоги до швидкодії [2]. Застосування технології GPGPU дозволяє отримати значний приріст швидкодії при роботі з зображеннями великої та надвеликої розмірності. Але через накладні витрати на організацію обчислень використання GPGPU для зображень малої та середньої розмірності не лише не дає приросту швидкодії, але й може сповільнити роботу системи [3]. Використання багатопоточності дозволяє одночасно працювати з декількома невеликими зображеннями, але не дає настільки значного приросту у випадку великих зображень [4].

Це зумовлює доцільність застосування концепції багаторівневого паралелізму до реалізації систем такого типу з метою отримання приросту швидкодії. Одним з методів, які доцільно реалізувати в межах концепції багаторівневого паралелізму є метод паралельно-ієрархічного (ПІ) перетворення завдяки його орієнтованості на паралельну реалізацію [5].

ПІ перетворення можливо визначити як принцип паралельного оброблення інформації, метою якого є досягнення максимально можливої алгоритмічної та схемотехнічної швидкодії при перетворенні інформації, зокрема зображень [5]. ПІ перетворення застосовується для виділення характерних ознак зображень, їх кодування і скорочення розмірності при виконанні обчислень. Добра

збіжність III перетворення використовується в структурах аналізу і розпізнавання зображень, при кодуванні і ущільненні даних, а також для обробки біомедичних сигналів.

Алгоритм III перетворення може бути ефективно реалізований на основі технології NVIDIA CUDA та бібліотеки CUDA Thrust [6]. Проте використання CUDA Thrust при реалізації паралельно-ієрархічного оброблення плямоподібних зображень на основі декількох GPU виявилось проблемним через технічні обмеження бібліотеки. Проблема полягає в тому, що майже всі функції цієї бібліотеки є синхронними, тобто блокують всі наступні операції до закінчення своєї роботи. Таким чином, нівелюються майже всі переваги Multi-GPU Programming, оскільки ця технологія розрахована в першу чергу на роботу з асинхронними функціями. Таким чином, при використанні синхронних функцій відеокарти починають працювати не паралельно, а по черзі, що призводить до фактичного перетворення паралельної програми у послідовну.

Вирішенням цієї проблеми стало комбіноване застосування технологій OpenMP та NVIDIA CUDA, тобто інтеграції CUDA-коду в багатопотокову структуру, що дозволило обійти обмеження бібліотеки CUDA Thrust. В результаті, у випадку одночасного використання двох графічних карт принцип роботи комбінованої програми прийняв наступний вигляд: спочатку в головному потоці здійснюється зчитування, попередня обробка та класифікація зображень та їх розподілення на два окремі набори, потім за допомогою засобів OpenMP на CPU викликаються два паралельних потоки, кожному з яких за допомогою засобів Multi-GPU Programming ставиться у відповідність своя графічна карта та свій набір зображень. В кожному з цих потоків здійснюється III оброблення відповідної групи зображень за допомогою обчислювальних потужностей відповідної графічної карти за тим самим принципом, що й в програмі на основі одного GPU. Таким чином, синхронність функцій бібліотеки CUDA Thrust перестає відігравати суттєву роль, оскільки кожен потік і кожна відеокарта працюють незалежно один від одного. По закінченню роботи усіх потоків отримані ними результати об'єднуються у головному потоці і виводяться користувачеві (рис. 1). Варто відзначити, що розроблена модель організації обчислень може бути застосована з довільною кількістю графічних карт за умови, що кількість графічних карт не перевищує кількості ядер CPU.

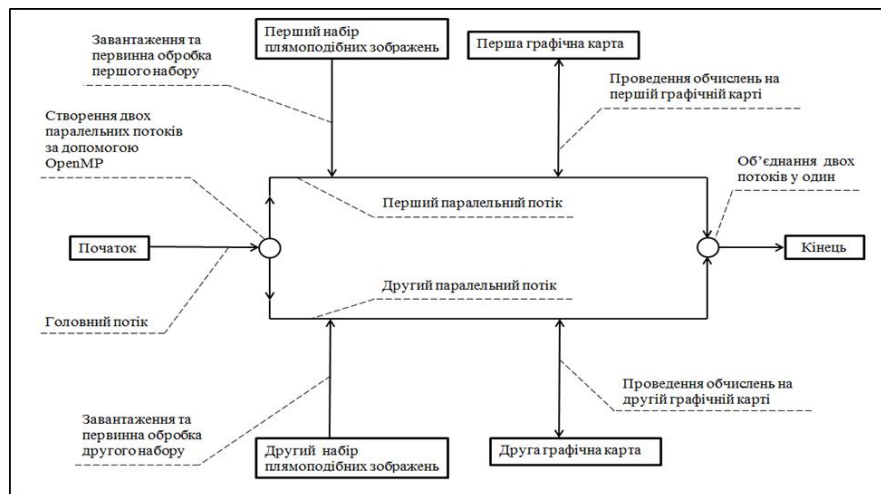


Рис. 1. Схема взаємодії NVIDIA CUDA та OpenMP при реалізації III перетворення

Однак не всі етапи процесу III оброблення зображень доцільно реалізовувати на основі GPU. Так, подібна реалізація модулів зчитування та попередньої класифікації зображень виявилась недоцільною через високі накладні витрати при організації обчислень. Проте використання технології OpenMP для одночасної роботи з декількома зображеннями в межах CPU за результатами експериментальних досліджень дозволило отримати приріст швидкодії в 1,22-2,75 разів (в залежності від розмірності зображення) при зчитуванні зображень та в 1,2-2,08 разів (в залежності від розмірності зображення) при попередній класифікації зображень.

Розроблений програмний продукт тестувався в чотирьох режимах:

- на основі використання CPU;
- на основі використання одного GPU;
- на основі використання двох GPU;

- на основі використання чотирьох GPU.

Обробленню підлягав набір з 50 зображень формату “.bmp” різної розмірності (128x128, 1024x1024, 2048x2048, 4096x4096 пікселів). Отримані результати наведені в таблиці 1.

Таблиця 1 – Результати роботи програмного продукту

Розмірність зображення (масиву)	Загальний час виконання програми,с			
	CPU	1 GPU	2 GPU	4 GPU
128x128	1,21808	1,56746	1,39699	1,26787
1024x1024	52,9144	47,6884	26,7139	17,4346
2048x2048	192,779	157,24	92,4078	63,4542
4096x4096	693,52	610,085	373,115	309,875

Висновки

Отримані результати показують доцільність і високу перспективність застосування концепції багаторівневого паралелізму для задач обробки та класифікації зображень, зокрема для реалізації ПП оброблення зображень. Так, розроблена модель організації обчислень дозволила отримати наступні результати: при розмірності зображень 1024x1024 пікселів за рахунок застосування 4 графічних карт досягається приріст швидкодії на 203% в порівнянні з послідовною реалізацією на CPU та на 173% в порівнянні з реалізацією на основі одного GPU, при розмірності зображень 2048x2048 пікселів досягається приріст швидкодії на 203% в порівнянні з послідовною реалізацією на CPU та на 147% в порівнянні з реалізацією на основі одного GPU, при розмірності зображень 4096x4096 пікселів досягається приріст швидкодії на 123% в порівнянні з послідовною реалізацією на CPU та на 96% в порівнянні з реалізацією на основі одного GPU.

Отримані результати можуть в подальшому бути застосовані для збільшення швидкодії процесу оброблення зображень різних типів, а не лише плямоподібних. Також створена модель організації обчислень може бути використана в процесі створення обчислювального кластеру на основі GPGPU – технологій.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Гергель В.П. Высокопроизводительные вычисления для многоядерных многопроцессорных систем. / Гергель В.П. – Н.: ННГУ им. Н.И.Лобачевского, 2010. – 421 с.
2. Форсайт Д. Компьютерное зрение. Современный подход / Форсайт Д., Понс Д. – М.: Мир, 2004. – 610с.
3. NVIDIA – WORLD LEADER IN VISUAL COMPUTING TECHNOLOGIES [Електронний Ресурс] – Режим Доступу: <http://www.nvidia.ru/page/home.html>.
4. Левін М. А. Параллельное программирование с использованием OpenMP [Електронний ресурс] / Левін М. А. // Режим доступу: <http://www.intuit.ru/studies/courses/1112/232/info>.
5. Яровий А. А. Методи та засоби організації високопродуктивних паралельно-ієрархічних обчислювальних систем із рекурсивною архітектурою : монографія / А. А. Яровий. – Вінниця: ВНТУ, 2016. – 363 с.
6. Яровий А.А. Обробка плямоподібних зображень методом прямого паралельно-ієрархічного перетворення на базі гетерогенних апаратних платформ / Яровий А.А., Кулик О.О. : Збірник тез доповідей X Міжнародної науково-практичної конференції ІОН-2016, (Вінниця, 11-14 жовтня, 2016 р.) - Вінниця : ВНТУ, 2016 – с. 9-11.

Кулик Олександр Олександрович — аспірант кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, Хмельницьке шосе, 95, e-mail: o.kulyk@vntu.edu.ua.

Яровий Андрій Анатольович — д.т.н., професор, професор кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, Хмельницьке шосе, 95, e-mail: a.yarovyy@vntu.edu.ua.

Olexandr O. Kulik — Postgraduate Student of Computer Science Department, Vinnytsia National Technical University, Vinnytsia, Khmelnytske Shose, 95, e-mail: o.kulyk@vntu.edu.ua.

Andrii A. Yarovyi — Doctor of Science (Eng.), Professor, Professor of Computer Science Department, Vinnytsia National Technical University, Vinnytsia, Khmelnytske shose, 95, e-mail: a.yarovyy@vntu.edu.ua.