УДК 681.3;004.93

**О.О. Кулик**
**А.А. Яровий**

# АНАЛІЗ СПОСОБІВ ВИКОРИСТАННЯ БАГАТОРІВНЕВОГО ПАРАЛЕЛІЗМУ У ЗАДАЧАХ ОБРОБКИ ТА КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

Вінницький національний технічний університет

***Анотація***
*Проведено аналіз способів використання багаторівневого паралелізму у задачах обробки та класифікації зображень, за результатами якого була підтверджена доцільність та перспективність комбінованого застосування існуючих технологій паралельних обчислень.*
**Ключові слова:** паралельні обчислення, GPGPU, паралельно-ієрархічне перетворення, багатопоточність.

***Abstract***
*The analysis of methods using multilevel parallelism in tasks of image processing and classification is carry out. The results of analysis confirmed the feasibility and expediency of the combination of existing technologies of parallel computing.*
**Keywords:** parallel computing, GPGPU, parallel-hierarchical transformation, multithreading.

## Вступ

Останнім часом спостерігається тенденція до зростання потреби у проведенні ресурсоємних обчислень, аналізу та обробки великих обсягів інформації. До задач такого типу відносяться, наприклад, задачі прогнозування погоди та змін клімату, моделювання складних фізичних процесів, розпізнавання та класифікації зображень, та багато інших [1]. Такий стан речей в свою чергу призводить до підвищення актуальності алгоритмів, які здатні використовувати усі переваги концепції паралелізму.

Метою роботи є аналіз способів використання багаторівневого паралелізму для підвищення швидкодії у задачах обробки та класифікації зображень.

## Результати дослідження

Під паралелізмом, як правило, розуміють концепцію обчислень, згідно з якою обрана задача розбивається на підзадачі, які виконуються одночасно. Паралелізм може бути реалізований у різних формах [2]:

- багатопоточність (виконання декількох потоків одночасно на одному CPU);
- обчислення на графічних картах (GPGPU), у тому числі на декількох GPU одночасно (Multi-GPU);
- паралельні обчислення на основі розподілених систем та інші.

Доцільність застосування кожної з них залежить від наявних ресурсів та поставленої задачі. Окрім того, певні форми паралелізму можуть використовуватися одночасно (багаторівневий паралелізм).

Під багатопоточністю зазвичай розуміють виконання декількох потоків одночасно на одному CPU. Робота багатопоточної програми починається з ініціалізації та виконання головного потоку (процесу), який у міру необхідності створює і виконує паралельні потоки, передаючи їм необхідні дані. Паралельні потоки з однієї паралельної області програми можуть виконуватися як незалежно один від одного, так і з пересилкою та отриманням повідомлень від інших паралельних потоків. Остання обставина ускладнює розробку програми, оскільки в цьому випадку програмісту доводиться займатися плануванням, організацією і синхронізацією посилки повідомлень між паралельними потоками.Структурно-функціональна організація багатопоточної програми зображена на рис. 1 [2].

На програмному рівні багатопоточність може бути реалізована за допомогою таких технологій як OpenMP, OpenCL, TBB або їх аналогів.

Прикладом задачі, в якій застосування OpenMP виявилось доцільним є задача класифікації плямоподібних зображень за заданими критеріями (розмір плями, ступінь її деформації, зміщення енергетичного центру та інші). В розглянутій програмнійреалізації набір зображень для класифікації розподілявся між декількома паралельними незалежними потоками, створеними за допомогою OpenMP, що дозволило досягти приросту швидкодії в 2,3 рази в порівнянні з послідовною схемою реалізації.

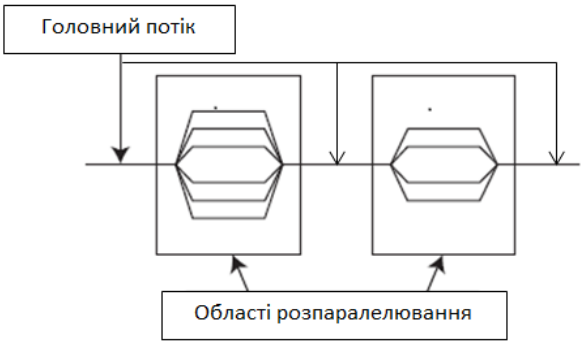


Рис. 1. Структурно-функціональна організація багатопоточної програми

Іншим підходом до реалізації паралелізму є GPGPU – технології. Під GPGPU розуміють техніку використання графічного процесору відеокарти для виконання неграфічних обчислень. Однією з головних переваг GPU є велика кількість ядер (близько 2500 в сучасних відеокартах), що дозволяє в повній мірі реалізувати переваги паралелізму. Теоретична потужність сучасних графічних карт вже набагато вища, ніж у відповідних CPU. Наразі існує велика кількість програмних засобів для реалізації GPGPU – технологій: NVIDIA CUDA, OpenCL, AMD FireStream та інші.

Найбільш перспективним з них є NVIDIA CUDA - це розроблена NVIDIA програмно-апаратна архітектура паралельних обчислень на основі мови програмування С[3]. В якості переваг CUDA варто відзначити широку підтримка паралелізму (в тому числі засобів Multi-GPU Programming), високий рівень інноватівності в порівнянні з аналогами та порівняно висока ефективність транзакцій між пам'яттю CPU та GPU. До недоліків CUDA варто віднести підтримку лише відеокарт від NVIDIA та неповну підтримку стандарту С, що в свою чергу накладає певні обмеження.

Окремо варто виділити Multi-GPU Programming це набір засобів, спрямованих на одночасне використання декількох графічних карт при виконанні CUDA-обчислень. Такий підхід забезпечує підвищення швидкодії та, з певними обмеженнями, збільшення розміру доступної для обчислень пам'яті. Окрім того, варто відзначити високу масштабованість даної технології. За потреби обчислювальна потужність Multi-GPU системи може бути збільшена шляхом підключення додаткових графічних карт, які можливо розмістити як на одному комп'ютері, так і на декількох, об'єднаних в одну мережу.

Варто відзначити, що технологію Multi-GPU Programming можливо реалізувати як на основі окремного комп'ютера з декількома GPU, так і на основі мережі з декількох комп'ютерів.Кожен з цих підходів має свої переваги та недоліки, але головним критерієм при виборі все ж залишається наявність потрібних апаратних засобів та відповідного програмного забезпечення.

Одним з напрямів, в яких використання GPGPU виявилось доцільним, є реалізація паралельно-ієрархічного перетворення (ПІ) [4]. В широкому сенсі ПІ перетворення можна розглядати як принцип паралельного оброблення інформації, орієнтований на досягнення максимально можливої алгоритмічної та схемотехнічної швидкодії. З практичної точки зору алгоритм ПІ перетворення складається з трьох основних операцій: транспонування, зсуву та G-перетворення. На вхід алгоритму подається матриця даних. В ході обробки за допомогою вищенаведених операцій знаходиться матриця хвостових елементів, яка і є результатом роботи алгоритму.

Алгоритм ПІ перетворення по своїй природі є орієнтованим на паралельне виконання, у тому числі на основі NVIDIA CUDA [5, 6]. Проте його реалізація на основі системи з двома GPU лише засобами Multi-GPU Programming виявилась недоцільною через синхронність ряду функцій.

Комбіноване застосування OpenMP та Multi-GPU Programming для створення двох незалежних потоків, кожен з яких працює зі своїм GPU та набором зображень дозволило обійти це обмеження та досягнути приросту швидкодії в 1,9 разів в порівнянні з використанням лише однієї графічної карти і майже в 106 разів в порівнянні з реалізацією на CPU. Варто відзначити, що найбільший приріст швидкодії реалізації на GPU в порівнянні з CPU досягається при обробці зображень великої та надвеликої розмірності (2048x2048 пікселів та більше).

Вищенаведене вказує на те, що комплексне застосування наявних програмних та апаратних засобів реалізації паралелізму дозволяє отримати значний приріст швидкодії навіть у складних та нетипових задачах і є перспективним для подальшого розвитку.

Так, комплексне застосування OpenMP та CUDA може виявитися доцільним в універсальній системі, яка буде обробляти зображення різної розмірності. Це зумовлено тим, що зображення малої розмірності, як правило, нераціонально обробляти на GPU, а OpenMP дає можливість опрацьовувати їх в декілька потоків на основі CPU. Окрім того, перспективним є об'єднання GPGPU - технологій та технологій розподілених обчислень з метою створення GPGPU-кластеру, який міг би об'єднати переваги обох наукових напрямів.

## Висновки

Отримані результати показують доцільність і високу перспективність комбінованого застосування технології паралельних обчислень для задач обробки та класифікації зображень. Так, застосування OpenMP в задачі класифікації плямоподібних зображень дозволило отримати приріст швидкодії в 2,3 рази в порівнянні з послідовною схемою реалізації. Комбіноване застосування OpenMP та Multi-GPU Programming при реалізації алгоритму ПІ перетворення на основі системи, що містить дві графічні карти, дозволило досягнути приросту швидкодії в 1,9 разів в порівнянні з використанням лише однієї графічної карти і майже в 106 разів в порівнянні з реалізацією на CPU.

Отримані результати планується використати в подальшій роботі з метою створення обчислювального кластеру на основі GPGPU – технологій.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ


1.Форсайт Д. Компьютерное зрение. Современный подход / Форсайт Д., Понс Д. – М.: Мир, 2004. – 610с.

2. Левін М. А. Параллельное программирование с использованием OpenMP [Електронний ресурс] / Левін М. А. // Режим доступу: http://www.intuit.ru/studies/courses/1112/232/info.

3. NVIDIA – WORLD LEADER IN VISUAL COMPUTING TECHNOLOGIES [ЕЛЕКТРОННИЙ РЕСУРС] – РЕЖИМ ДОСТУПУ: HTTP://WWW.NVIDIA.RU/PAGE/HOME.HTML.

4. Паралельно-ієрархічне перетворення як системна модель оптико-електронних засобів штучного інтелекту : [Монографія.] / В.П. Кожем'яко, Ю.Ф. Кутаєв, С.В. Свєчніков, Л.І. Тимченко, А.А. Яровий – Вінниця: УНІВЕРСУМ-Вінниця, 2003. – 324 с.

5. Перспективи застосування технології NVIDIA SLI для паралельно-ієрархічної обробки зображень / Яровий А.А., Кулик О.О. : Збірник тез доповідей VII Міжнародної науково-технічної конференції [Фотоніка ОДС-2015)], (Вінниця, 21-23 квітня 2015 р.) – Вінниця, ВНТУ, 2015. – с. 10.

6. Яровий А. А. Паралельно-ієрархічне перетворення плямоподібних зображень на основі Multi-GPU систем / А. А. Яровий, О. О. Кулик, Н. І. Кокряцька // Інформаційні технології та комп'ютерна інженерія. – 2015. – №3(34). – С. 72-80.



**Кулик Олександр Олександрович —** магістрант кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, Хмельницьке шосе, 95, e-mail: alex.kaiser13@mail.ru.

**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@gmail.com.

**Olexandr O. Kulyk —** Graduate Student of Computer Science Department, Vinnytsia National Technical University, Vinnytsia, Khmelnytske Shose, 95, e-mail: alex.kaiser13@mail.ru.

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