



Матеріали VI Міжнародної науково-практичної конференції молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології» (SoftTech-2024)



2021



19-21 травня
Україна, Київ

ISSN 2306-7233 (Online)

ISSN 2306-6962 (Print)

М А Т Е Р І А Л И

**VI Міжнародної науково-практичної конференції молодих вчених та студентів
«ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
І ПЕРЕДОВІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ (SOFT TECH-2024)»
21-23 травня 2024 року, м.Київ**

Інженерія програмного забезпечення і передові інформаційні технології (Soft Tech-2024): матеріали VI Міжнародної науково-практичної конференції молодих вчених та студентів, 21-23 травня 2024 року, м. Київ, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», ФІОТ, 167 с.

Збірник містить тези доповідей, що були представлені на міжнародній науково-практичній конференції молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології (SoftTech-2024)». В доповідях розглянуті сучасні наукові та практичні проблеми інформатики та програмної інженерії.

Матеріали подано в авторській редакції

Дизайн обкладинки

Майер З.О., провідний інженер кафедри інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського»

ЗМІСТ

Безотосний Данііл Костянтинівич, Сидоров Микола Олександрович ПРЕДСТАВЛЕННЯ ЕКОСИСТЕМ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА СТРУКТУРНОМУ РІВНІ.....	8
Гаврильченко Олег Олегович, Баклан Ігор Всеволодович ПРОГРАМНІ ТА МОВНІ ЗАСОБИ ДЛЯ КОМПЛЕКСА РАНІШНЬОЇ ДІАГНОСТИКИ ХРОНІЧНИХ НЕВРОЛОГІЧНИХ ЗАХВОРЮВАНЬ.....	12
Галушко Дмитро Олександрович КОНЦЕПЦІЯ ОПТИМІЗАЦІЇ ПРОЦЕСІВ ВИРОБНИЦТВА ПРОДУКТІВ ДЛЯ ПРОВАЙДЕРІВ ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНИХ ПОСЛУГ.....	16
Гобов Денис Андрійович ПРОБЛЕМИ З ЯКІСТЮ ВИМОГ В ІТ-ПРОЕКТАХ.....	22
Голяченко Анастасія Миколаївна, Люшенко Леся Анатоліївна РОЗРОБКА ПРОГРАМНОЇ ЕКСПЕРТНОЇ СИСТЕМИ ВИБОРУ АЛГОРИТМІВ ПРОГНОЗУВАННЯ КРИПТОВАЛЮТИ У РЕЖИМІ РЕАЛЬНОГО ЧАСУ.....	26
Губський Андрій Миколайович МЕХАНІЗМ САМОДІАГНОСТИКИ ВИСОКОНАВАНТАЖЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМ.....	32
Добрянський Богдан Ігорович, Стеценко Інна Вячеславівна УНІФІКОВАНИЙ ІНТЕРФЕЙС ДЛЯ ВЗАЄМОДІЇ З БЛОКЧЕЙН МЕРЕЖАМИ.....	37
Довгополюк Роман Русланович, Олійник Юрій Олександрович МЕТОД ГЕНЕРАЦІЇ ТЕКСТІВ УКРАЇНСЬКОЮ МОВОЮ.....	41
Дубовик Андрій Павлович, Фіногенов Олексій Дмитрович ОСОБЛИВОСТІ ГЕНЕРУВАННЯ КОДУ МОДУЛЬНИХ ТЕСТІВ GUI ANDROID-ДОДАТКІВ, РОЗРОБЛЕНИХ З ВИКОРИСТАННЯМ ДЕКЛАРАТИВНОЇ ПАРАДИГМИ.....	46
Завальнюк Євген Костянтинівич, Романюк Олександр Никифорович ТЕНДЕНЦІЇ РОЗВИТКУ ПРОГРАМНИХ ЗАСОБІВ КОМП'ЮТЕРНОЇ ГРАФІКИ.....	53
Зарічковий Олександр Анатолійович, Стеценко Інна Вячеславівна ПОКРАЩЕННЯ КРОСС-МОДАЛЬНИХ ДОСЛІДЖЕНЬ ЗНАНЬ VLM.....	58
Капшук Марія Валеріївна, Олійник Юрій Олександрович ВИЯВЛЕННЯ АНОМАЛІЙ НА МРТ-ЗНІМКАХ.....	62
Коровін Данііл Дмитрович, Божуха Лілія Миколаївна ПРАКТИЧНА КОРИСТЬ ВИКОРИСТАННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ У РОЗРОБЛЕННІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	69
Кришталь Віктор Олегович, Сидоров Микола Олександрович МЕТОД НАВЧАННЯ З ПІДКРІПЛЕННЯМ ДЛЯ УПРАВЛІННЯ ВИТРАТАМИ НА ОПАЛЕННЯ.....	72

Теленик Сергій Федорович, Гжегош Новаковський, Гавриленко Олена Валеріївна, Чимшир В'ячеслав Іванович, Омельченко Ростислав Володимирович	
ДОСЛІДЖЕННЯ ПРОЦЕСІВ, ПОВ'ЯЗАНИХ З ПРОЕКТУВАННЯМ СЕРВІСІВ У ІНФОРМАЦІЙНИХ СИСТЕМАХ ПРОВАЙДЕРІВ.....	140
Форкерт Павло Павлович, Сидорова Марина Геннадіївна	
ПОКРАЩЕННЯ ПЕРЕЛІКІВ У ДІАЛЕКТІ МОВИ ПРОГРАМУВАННЯ GO.....	148
Форосяний Артур Богданович, Романюк Олександр Никифорович	
ВИКОРИСТАННЯ ВІДЕОКАРТ ДЛЯ НЕГРАФІЧНИХ РОЗРАХУНКІВ.....	151
Хусаїнов Дмитро Павлович, Хусаїнов Павло Валентинович	
ДОВЕДЕННЯ ПРАВИЛЬНОСТІ ПРОГРАМИ ПРОТИ MEMORY CORRUPTION.....	155
Цуканова Марина Сергіївна, Ліхоузова Тетяна Анатоліївна	
РОЗРОБКА ПОРТАЛУ СЛУЖБИ ПІДТРИМКИ ІТ ВІДДІЛУ.....	158
Шпилька Владислав Сергійович, Стеценко Інна Вячеславівна	
МЕТОДИ ОДНОКЛАСОВОЇ КЛАСИФІКАЦІЇ ТА ВИЯВЛЕННЯ СПІЛЬНОГО ОБ'ЄКТУ НА ЗОБРАЖЕННЯХ	163

Форостяний Артур Богданович, студент 1 курсу спеціальності
«Інженерія програмного забезпечення»

Вінницький національний технічний університет, Україна

Науковий керівник: Романюк Олександр Никифорович, доктор технічних наук,
професор кафедри програмного забезпечення

Вінницький національний технічний університет, Україна

ВИКОРИСТАННЯ ВІДЕОКАРТ ДЛЯ НЕГРАФІЧНИХ РОЗРАХУНКІВ

Анотація. Розглянуто використання графічних відеокарт для паралельних і розподілених обчислень. Оцінено ефективність виконання даних задач.

Ключові слова: графічна відеокарта; паралельні обчислення; CUDA; OpenCL; машинне навчання.

Forostianyi Artur, 1st year student, "Software Engineering"

Vinnitsia National Technical University, Ukraine

Academic supervisor: Romaniuk Oleksandr, Doctor of Technical Sciences,
Professor, Head of the Software Department

Vinnitsia National Technical University, Ukraine

USE OF VIDEO CARDS FOR NON-GRAPHIC CALCULATIONS

Abstract. The use of graphics cards for parallel and distributed computing is considered. The efficiency of these tasks is evaluated.

Keywords: graphics video card; parallel computing; CUDA; OpenCL; machine learning.

Основна частина. Графічна відеокарта [1] – це пристрій для прискореного відображення графічної інформації (GPU). Особливість GPU полягає у використанні багатьох ядер. Так, зокрема, відео карта RTX 3060 SUPER має 5632 ядер CUDA. Сьогодні графічна продуктивність суттєво випереджає продуктивність центральних процесорів. Це обумовлює доцільність використання графічних відеокарт для розв'язку багатьох ресурсомістких задач. Використання відеокарт для неграфічних розрахунків, або GPU-обчислення, є технологічною тенденцією останніх років. Замість того, щоб обмежувати їхнє використання виключно для графіки, сучасні відеокарти стають потужними інструментами для широкого спектру обчислювальних завдань, що вимагають великої продуктивності та паралельних обчислень. GPU-обчислення, або обчислення загального призначення на графічних процесорах (GPGPU) [2] – це процес використання графічного процесора (GPU) для виконання обчислень в додатках, що, зазвичай, виконуються центральним процесором (CPU) [3]. На перший погляд здається, що використання GPU замість CPU є недоцільним, адже перший призначений для графіки, а не для неграфічних задач. На практиці виявляється, що використання декількох відеокарт в одному комп'ютері або великої кількості графічних чипів децентралізує паралельну природу обробки графіки. Ще в 80-х роках XX століття, задовго до появи компаній по виготовленню чипів для відеокарт, для неграфічних обчислень використовували трансп'ютери [4] – елементи побудови багатопроцесорних систем, що спочатку

використовувались для виконання даних задач. З появою відеокарт трансп'ютери втратили свою актуальність, тим не менш вони все ще використовуються у вигляді спеціалізованих прискорювачів. Згодом на ринку з'являються такі компанії як Nvidia та AMD (ATI) – лідери по виробництву графічних чіпів, що використовуються в усіх відеокартах сьогодні. Для використання графічних процесорів в неграфічних обчисленнях слід використовувати відповідні для цього технології. Найвідомішими з них є CUDA та OpenCL. CUDA [5] (Compute Unified Device Architecture) – програмно-апаратна архітектура паралельних обчислень, що розроблена та підтримується компанією Nvidia. Для написання програм використовують спрощену мову C. На даний час ця технологія присутня в графічних процесорах таких лінійок відеокарт Nvidia як GeForce, Quadro та Tesla. Ще один схожий фреймворк – OpenCL [6] (Open Computing Language). Він теж призначений для створення комп'ютерних програм, пов'язаних з паралельними обчисленнями на різних графічних і центральних процесорах. Дана технологія підтримується та використовується такими компаніями, як Apple, AMD, ARM, Qualcomm та ін. Спектр використання GPU-обчислень є доволі широким. Графічні чіпи оптимізовані для виконання ресурсомісткої роботи завдяки моделі SIMT [7], в основі якої ідея розбиття обчислювального завдання на невеликі фрагменти, які виконуються паралельно на кожному ядрі графічного процесора. Такий підхід виконання можна використовувати для математичних симуляцій, наприклад, для підсумовування багатьох точок на еліптичній кривій, розв'язування диференціальних рівнянь, сортування елементів в масиві тощо. Ще один приклад такого обчислення – наукове моделювання. Наприклад, моделювання [8] обчислювальної гідродинаміки, що використовується в таких галузях, як аерокосмічна та автомобільна техніка, вимагає високого ступеня паралельної обробки. Наукові дослідження в таких галузях, як астрофізика, прогнозування погоди та моделювання згортання білків, також виграли від використання графічних обчислень: час, необхідний для проведення моделювання, скоротився з тижнів або місяців до декількох днів або годин. Також слід згадати про машинне навчання – процес, завдяки якому розвиваються різні моделі штучного інтелекту. Обчислювальна потужність відеокарт дозволяє обробляти паралельні обчислення з високою швидкістю. Щоправда не всі відеокарти здатні на це: для таких операцій потрібні потужніші варіанти, наприклад, з лінійки Nvidia Tesla [9], які містять тензорні ядра – високопродуктивні графічні ядра, чия потужність досягається завдяки динамічній оптимізації обчислень. Саме вони здатні прискорити навчання штучного інтелекту в рази, що є важливим для розвитку ШІ.

Висновки. GPU-обчислення стає все більш значущою технологією, яка дозволяє використовувати відеокарти для неграфічних завдань. Ця практика дозволяє виконувати широкий спектр обчислювальних завдань, що вимагають великої продуктивності та паралельних обчислень. Впровадження GPU-обчислень в наукові дослідження, машинне навчання, моделювання та інші галузі дозволяє значно збільшити швидкість та ефективність обробки даних. Однак, для найбільш вимогливих завдань потрібні спеціалізовані відеокарти з тензорними ядрами, що підтримують динамічну оптимізацію обчислень. Розвиток GPU-обчислень відкриває нові перспективи для подальшого росту обчислювальних можливостей та розширення їх застосувань у різних сферах.

Література

1. Відеокарта [Електронний ресурс] / Wikipedia. – Режим доступу <https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%B5%D0%BE%D0%BA%D0%B0%D1%80%D1%82%D0%B0> – Назва з екрану.
2. GPGPU [Електронний ресурс] / Wikipedia. – Режим доступу <https://uk.wikipedia.org/wiki/GPGPU> – Назва з екрану.
3. Центральний процесор [Електронний ресурс] / Wikipedia. – Режим доступу https://uk.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BD%D1%82%D1%80%D0%B0%D0%B%D1%8C%D0%BD%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80 – Назва з екрану.
4. Трансп'ютер [Електронний ресурс] / Wikipedia. – Режим доступу <https://uk.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80> – Назва з екрану.
5. CUDA [Електронний ресурс] / Wikipedia. – Режим доступу <https://uk.wikipedia.org/wiki/CUDA> – Назва з екрану.
6. OpenCL [Електронний ресурс] / Wikipedia. – Режим доступу <https://uk.wikipedia.org/wiki/OpenCL> – Назва з екрану.
7. Single instruction, multiple threads [Електронний ресурс] / Wikipedia. – Режим доступу https://en.wikipedia.org/wiki/Single_instruction,_multiple_threads – Назва з екрану.
8. Обчислення на GPU: сфери застосування [Електронний ресурс] / Wikipedia. – Режим доступу <https://introserv.com/ua/blog/obchislennya-na-gpu-sferi-zastosuvannya/> – Назва з екрану.
9. Nvidia Tesla [Електронний ресурс] / Wikipedia. – Режим доступу https://uk.wikipedia.org/wiki/Nvidia_Tesla – Назва з екрану.

References

1. Videocard [Electronic resource] / Wikipedia. – Access mode <https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%B5%D0%BE%D0%BA%D0%B0%D1%80%D1%82%D0%B0> – Title from screen.
2. GPGPU [Electronic resource] / Wikipedia. – Access mode <https://uk.wikipedia.org/wiki/GPGPU> – Title from screen.
3. Central processing unit [Electronic resource] / Wikipedia. – Access mode https://uk.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BD%D1%82%D1%80%D0%B0%D0%B%D1%8C%D0%BD%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80 – Title from screen.
4. Transputer [Electronic resource] / Wikipedia. – Access mode <https://uk.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80> – Title from screen.
5. CUDA [Electronic resource] / Wikipedia. – Access mode <https://uk.wikipedia.org/wiki/CUDA> – Title from screen.
6. OpenCL [Electronic resource] / Wikipedia. – Access mode <https://uk.wikipedia.org/wiki/OpenCL> – Title from screen.
7. Single instruction, multiple threads [Electronic resource] / Wikipedia. – Access mode https://en.wikipedia.org/wiki/Single_instruction,_multiple_threads – Title from screen.

8. GPU computation: spheres of computation [Electronic resource] / Wikipedia. – Access mode <https://introserv.com/ua/blog/obchislennya-na-gpu-sferi-zastosuvannya/> – Title from screen.

9. Nvidia Tesla [Electronic resource] / Wikipedia. – Access mode https://uk.wikipedia.org/wiki/Nvidia_Tesla – Title from screen.