

## АНАЛІЗ ЕФЕКТИВНОСТІ РОЗПАРАЛЕЛЕННЯ ОБЧИСЛЕНЬ НА GPU З ВИКОРИСТАННЯМ РІЗНИХ АРХІТЕКТУР

*У статті проведено порівняльний аналіз відеоадаптерів з різними архітектурами – Tesla, Fermi, Kepler. Проаналізовано особливості новітньої архітектури Kepler у порівнянні з попередніми. Проведено чисельні експерименти розпаралелення на GPU з використанням різних архітектур.*

*Ключові слова: графічний процесор, GPU, розпаралелення, паралельні обчислення, Kepler, Fermi.*

I.P. STRUBYTSKA

Ternopil National Economic University

### ANALYSIS OF EFFICIENCY OF PARALLIZATION OF COMPUTING ON GPU WITH USING DIFFERENT ARCHITECTURES

*The comparative analysis of video adapters with different architectures (Tesla, Fermi, Kepler) are conducted in this article. The features of the modern architecture of Kepler are compared to previous ones. Numerical experiments of parallelization on GPU using different architectures are conducted.*

*Keywords: graphics processor, GPU, parallelization, parallel computing, Kepler, Fermi.*

#### Вступ

Ріст частот універсальних процесорів зупиняють фізичні обмеження і високе енергоспоживання. Збільшення їх продуктивності все частіше відбувається за рахунок розміщення декількох ядер в одному процесорі. Присутні зараз на ринку процесори містять лише до чотирьох ядер (подальше зростання не буде швидким) і призначені для звичайних програм, які використовують архітектуру MIMD. Кожне ядро працює окремо від інших, виконуючи різні інструкції для різних процесів.

Спеціалізовані векторні можливості появились в універсальних процесорах, в першу чергу, через високі вимоги графічних програм. Саме тому для певних задач (виконання однотипних дій з різними даними) застосування GPU (Graphics Processing Unit) вигідніше ніж CPU (Central Processing Unit).

Сучасні GPU - це багатоядерні системи SIMD-архітектури з достатньо високою (до 1 Тфлопс) піковою продуктивністю. Порівняно з традиційними архітектурами, вони мають порівняно низьку характеристику «ціна/продуктивність», що викликає зацікавлення використовувати GPU не тільки для обробки графічної інформації, але й для вирішення будь-яких обчислювальних задач [1].

На сьогодні на ринку графічні процесори, як окремі компоненти персональних комп'ютерів, випускають дві компанії: NVIDIA і AMD. Є ще Intel, проте вона спеціалізується на випуску GPU для вбудованих відеокарт. Тому при розгляді графічних процесорів для використання їх у обчисленнях будемо використовувати продукцію цих двох компаній.

AMD FireStream — це потоковий процесор, розроблений компанією AMD, призначений для збільшення ефективності розв'язку задач, з високим ступенем паралелізму. Обчислювальні прискорювачі FireStream доступні в системах усіх провідних виробників серверів і можуть використовуватись в масштабуючих серверах, блейд-серверах.

GeForce — сімейство GPU і чіпсетів материнських плат компанії NVIDIA, яке орієнтується на споживацький ринок. GeForce переважно використовується у відеоадаптерах для персональних і переносних комп'ютерів.

Quadro — це лінія відеоадаптерів для професійних дизайнерів, які надають багато можливостей для обробки зображень та відео [2].

Tegra — сімейство GPU, яке використовується в мобільних рішеннях (мобільних телефонах, смартфонах, планшетах).

Tesla — це сімейство обчислювальних систем NVIDIA, які можна використовувати для наукових і технічних обчислень загального призначення. Tesla не може повністю замінити звичайний універсальний процесор, але дозволяє використовувати обчислювальний ресурс множини своїх ядер для розв'язку ресурсомістких задач. Перевагами цих процесорів є велика енергоефективність, недоліком — менша універсальність [3].

Сьогодні компанія NVIDIA пропонує підтримку обчислень на всіх рівнях: апаратному (універсальні процесори GPU, висока швидкість обміну даними), драйверному (використання універсальних механізмів не прив'язаних до конкретних технологій), користувацькому (розробка бібліотек, компіляторів і SDK з прикладами програм і документацією) [4].

На даний час обчислення на графічних процесорах з технологією CUDA — це інноваційне поєднання обчислювальних особливостей нового покоління графічних процесорів NVIDIA, що обробляють відразу тисячі потоків з високим рівнем інформаційного завантаження, які доступні через стандартну мову програмування C [5].

### Порівняльний аналіз різних архітектур графічних процесорів

У праці [6] порівняно часову складність виконання програми на різних графічних процесорах. Швидкодія виконання залежить від кількості ядер графічного процесора. З швидким розвитком багатоядерних процесорів змінюється і їх архітектура, що якісно впливає на продуктивність обчислень.

Для позначення можливостей GPU CUDA використовують поняття Compute Capability, яке позначається парою цілих чисел: major.minor. Перше число позначає глобальну архітектурну версію, друге – модифікацію.

На сьогодні існують такі CUDA Compute Capability:

1. Покоління Tesla (не слід плутати з лінією продуктів для HPC):
  - 1.1 – базові можливості CUDA, атомарні операції з глобальною пам'яттю;
  - 1.2 – атомарні операції зі спільною пам'яттю, warp vote-функції;
  - 1.3 – обчислення з подвійною точністю;
2. Покоління Fermi:
  - 2.0 – нова архітектура чіпа, асинхронне виконання ядер;
  - 2.1 – нова архітектура warp scheduler-ів;

#### 3. Покоління Kepler:

- 3.0, 3.2 – нова архітектура чіпа, Unified memory programming;
- 3.0 – динамічний паралелізм, Hyper Queue;

#### 4. Покоління Maxwell:

- sm\_50 and sm\_52 – нова архітектура чіпа.

Для ефективного програмування з використанням GPU потрібно враховувати Compute Capability пристрою, що використовується. Проведемо порівняння трьох архітектур: Tesla, Fermi та Kepler.

«Одиницею»

побудови пристрою графічного процесора (як ядро в CPU) є потоковий мультипроцесор (Streaming Multiprocessor, SM). Мультипроцесор об'єднує основні обчислювальні потужності GPU: текстурні блоки, геометричний двигун PolyMorph Engine і масив ядер CUDA.

Кожне ядро CUDA представляє собою повністю конвейеризований процесор з одним цілочисельним ALU і блоком обчислень з плаваючою комою. За допомогою сотень таких ядер GPU виконує шейдерні програми та обчислення для неграфічних додатків з API OpenCL, DirectCompute, PhysX і, власне, CUDA API.

В середині SM ядра CUDA (рис. 1) використовується спільно з іншими обчислювальними компонентами: блоками Load/Store (LD/ST),

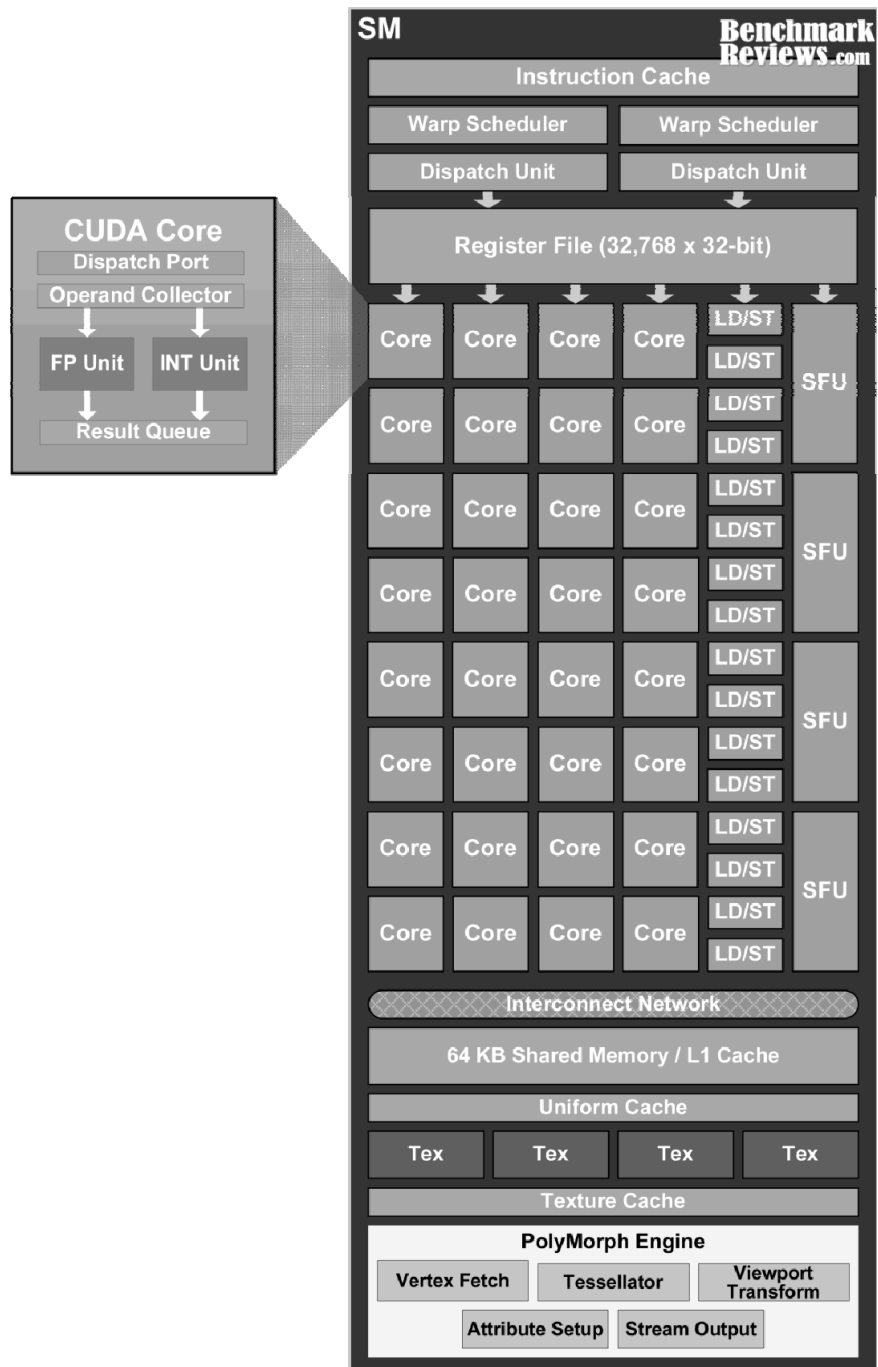


Рис. 1. Потоковий мультипроцесор і ядро CUDA [8]

Інформаційно-вимірювальні та обчислювальні системи і комплекси в технологічних процесах

текстурними блоками, блоками інтерполяції, блоками обчислення спеціальних функцій (Special Function Units, SFU). Всі ці компоненти отримують інструкції для виконання від одних і тих же диспетчерів [7].

У Fermi наявні [8]:

- 32 скалярних ядра CUDA Core, ~1.5ГГц;
- 2 Warp Scheduler;
- файл реєстрів, 128KB;
- 3 кеша – текстурний, глобальний (L1), константний (uniform);
- PolyMorphEngine – графічний конвеєр;
- текстурні юніти;
- 16 x Special Function Unit (SFU) – інтерполяція і трансцендентна математика одинарної точності
- 16 x Load/Store

У Fermi чіп з максимальною конфігурацією має 16 SM, що рівне 512 ядрам CUDA.

Архітектура Fermi представлена на рис. 2.

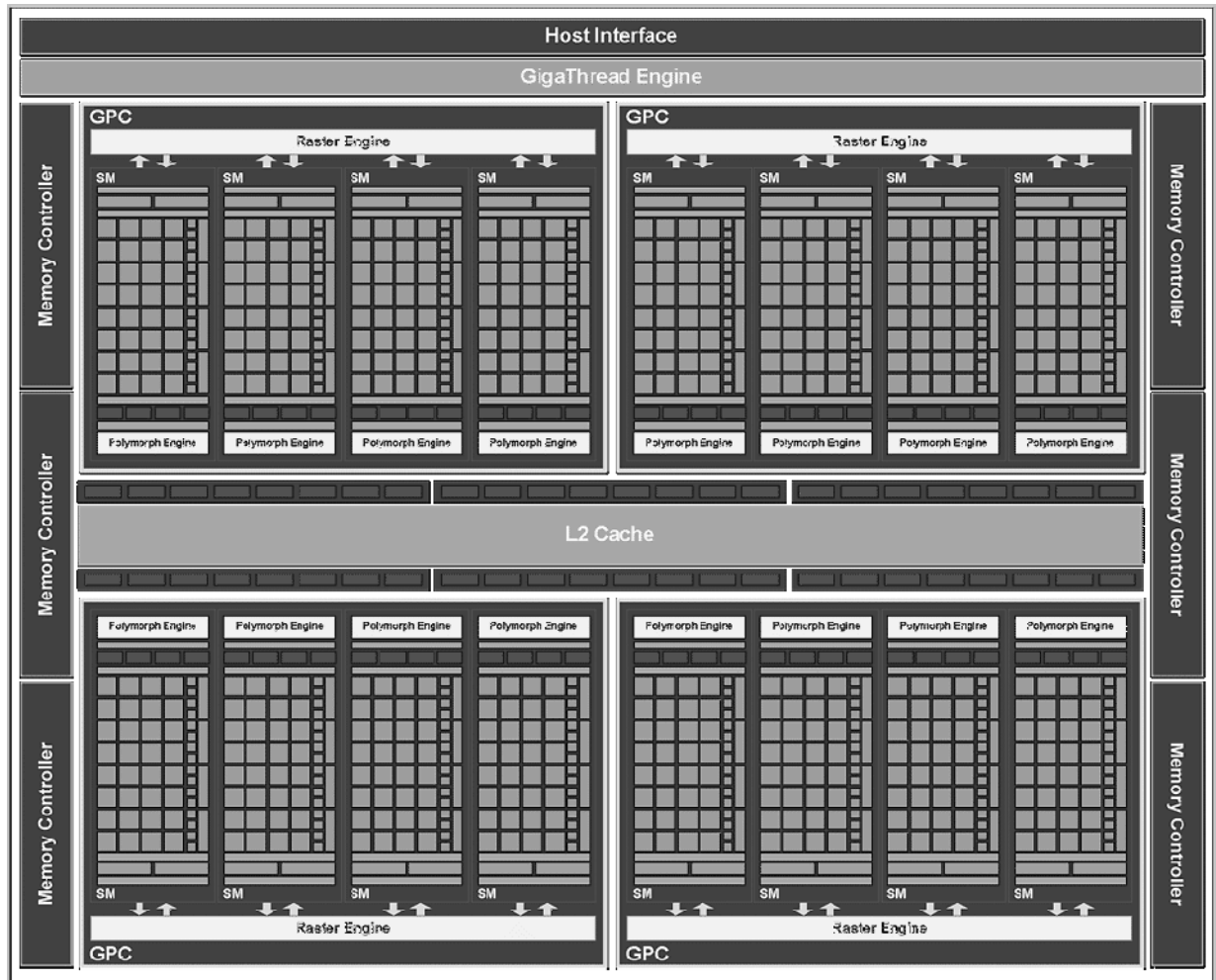


Рис. 2. Архітектура Fermi [8]

У Kepler потокові мультипроцесори були дуже перероблені. Тому до їх абревіатури добавили літеру X (SMX). Як і у Fermi, мультипроцесор об'єднує основні обчислювальні потужності графічного процесора: текстурні блоки, геометричний двигун PolyMorph Engine і масив ядер CUDA.

У Kepler всі блоки SMX (рис. 3) працюють на одній частоті і їх стало більше. Конфігурація мультипроцесора наступна[8]:

- 192 ядра CUDA;
- 2 Warp Scheduler;
- 32 блоків інтерполяції;
- 32 блоків LD/ST;
- 64 x DP Unit;
- 32 x SFU;
- 256KB реєстрів.

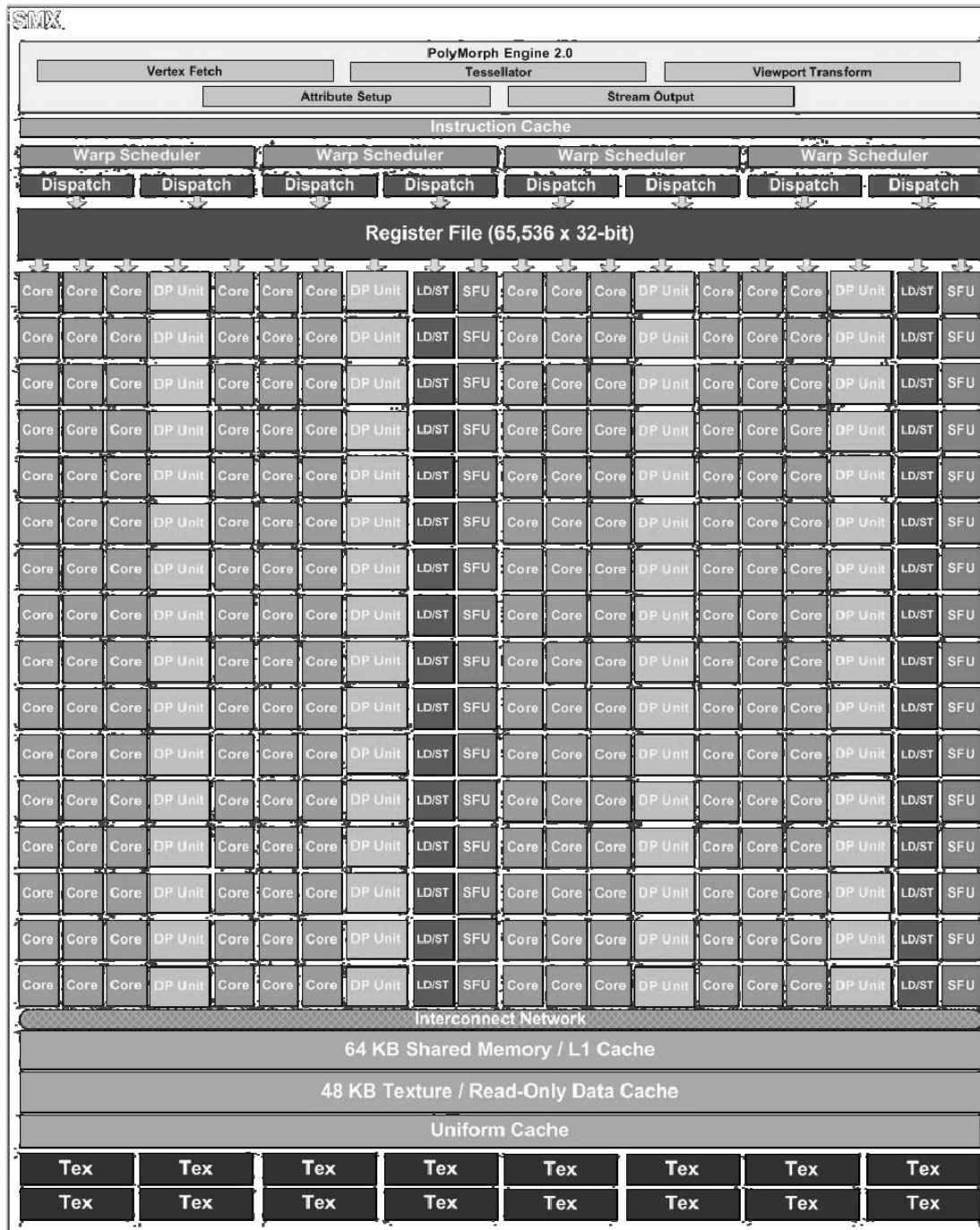


Рис. 3. Поточковий мультипроцесор у Kepler [8]

У склад SMX входить ще один блок з восьми ядер CUDA, який скритий на діаграмі. Це спеціальні ядра, які можуть виконувати обчислення з подвійною точністю (FP64) [7].

У порівнянні з Fermi, у Kepler стало в два рази більше планувальників. З кожним планувальником зв'язані два диспетчери. Вони можуть одночасно відправляти на виконання зразу два «ряди» інструкцій з одного warp. Таким чином, поточковий мультипроцесор набуває функцію позачергового виконання.

У Fermi планувальник визначав залежності операцій у шейдерному коді та перевпорядковував виконання різних warp. У Kepler задача залежностей покладена на компілятор. У самій інструкції вказується, на якому етапі в майбутньому вона може бути відправлена на виконання, і поки цей момент не наступив, планувальник вибирає для виконання інші warp. З одного боку складні планувальники обтяжують енергетичний бюджет, але з іншого – ефективність неграфічних обчислень без них постраждає. Тобто NVIDIA в архітектурі Kepler пожертвувала продуктивністю в користь енергетичної ефективності.

Як і процесори Fermi, Kepler (рис. 4) має легко масштабований модульний дизайн. Всі обчислювальні компоненти розподілені між чотирма «графічними кластерами» (Graphics Processing Cluster, GPC). Поза кластерами знаходиться тільки загальний кеш L2, контролери пам'яті, ROP і блок GigaThread Engine, який розподіляє навантаження між GPC [7].

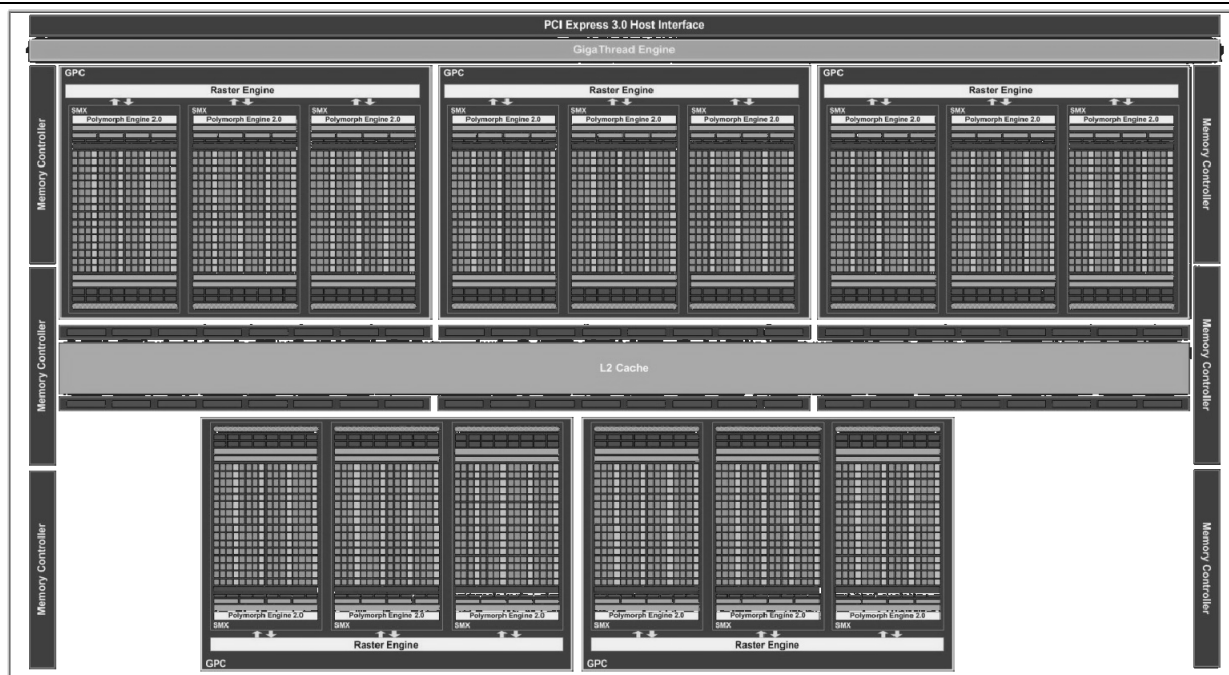


Рис. 4. Архітектура Kepler [8]

У Kepler чіп в максимальній конфігурації має 15 SMX, тобто 2880 cuda-ядра.

Отже, основними особливостями новітньої архітектури Kepler, у порівнянні з попередніми є [9, 10]:

1. Висока продуктивність і ефективність досягається у SMX шляхом збільшення процесорних ядер і зменшуючи логіку управління (рис. 5).

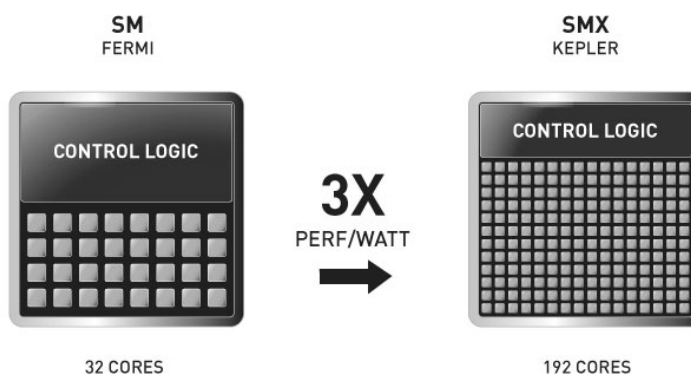


Рис. 5. Порівняння будови потокового мультипроцесора Fermi та Kepler [9]

2. Динамічний паралелізм на Kepler породжує нові потоки шляхом адаптації до даних без повернення до CPU (рис. 6). Це значно спрощує програмування GPU і прискорює більший набір популярних алгоритмів.

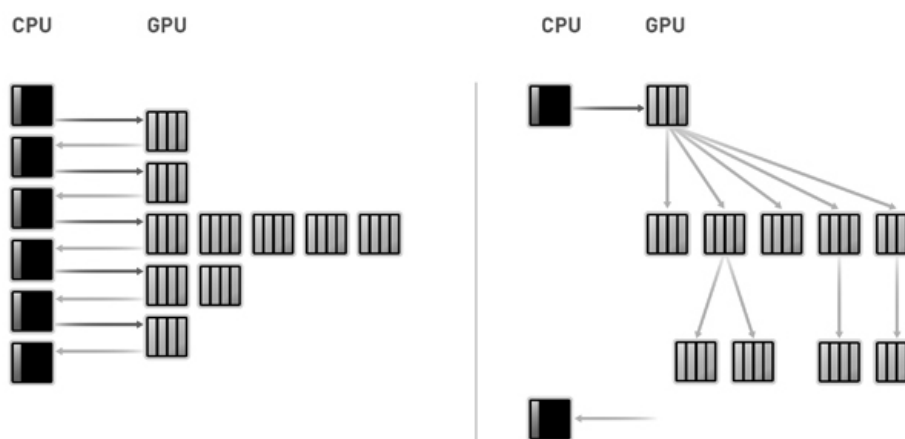


Рис. 6. Порівняння динамічного паралелізму у Fermi та Kepler [9]

3. Нурер-Q у Kepler підвищує ефективність використання GPU, забезпечуючи доступ потоків до 32 незалежних апаратних черг роботи. Нурер-Q дозволяє декільком ядрам CPU запускати роботу на одному GPU одночасно. Це значно збільшує ефективність використання GPU і зменшує час простою CPU.

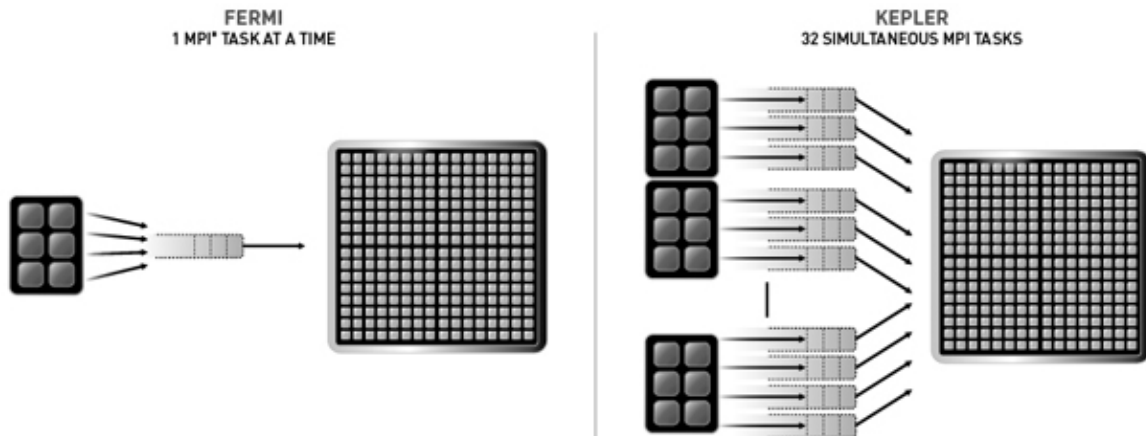


Рис. 7. Порівняння Нурер-Q у Fermi та Kepler [9]

#### Оцінка ефективності розпаралелення на GPU з використанням різних архітектур

У праці [11] запропоновано побудувати дискретну динамічну модель двообмоткового трансформатора з використанням розпаралелення на графічному багатоядерному процесорі. Вхідними величини є напруги на первинній і вторинній обмотках, а вихідними – відповідні сили струму. Перехідні характеристики трансформатора було знято експериментально. Спершу вимірювали сили струму в обох обмотках при поданні на первинну обмотку стрибка напруги при закороченій вторинній обмотці. Тоді вимірювали сили струму при стрибку напруги на вторинній і закороченій первинній обмотках. Частота дискретизації вимірювань складала 8 кГц. Для існуючого набору експериментальних даних було побудовано лінійну неавтономну модель двообмоткового трансформатора:

$$\begin{cases} \vec{x}^{(k+1)} = \begin{pmatrix} 0,9316 & 0,4713 \\ 0,0053 & 0,9343 \end{pmatrix} * \vec{x}^{(k)} + \begin{pmatrix} 0,0111 & 0,0101 \\ -0,0011 & 0,0008 \end{pmatrix} * \vec{v}^{(k)} \\ \vec{y}^{(k+1)} = \begin{pmatrix} 0,0018 & 0,0171 \\ 0,0011 & -0,0125 \end{pmatrix} * \vec{x}^{(k+1)} + \begin{pmatrix} 0,0128 & 0,0008 \\ 0,0004 & 0,0078 \end{pmatrix} * \vec{v}^{(k)} \end{cases}$$

де  $\vec{x}^{(k)}$  – вектор змінних стану, який характеризує поточний стан об'єкту;

$\vec{v}^{(k)}$  – вектор вхідних значень;

$\vec{y}^{(k)}$  – вектор вихідних значень.

При побудові дискретних динамічних моделей функція мети має чітко виражений яровий характер з великою кількістю локальних мінімумів. Для розв'язку такої задачі використано метод напрямного конуса Растрігіна. Проте така оптимізаційна задача є досить складною і вимагає значних обчислювальних затрат, які зумовлюють високі вимоги до швидкодії та необхідної оперативної пам'яті обчислювальних засобів. Тому доцільним є розпаралелення обчислювального процесу побудови дискретних динамічних моделей.

Враховуючи те, що в цій задачі виконується велика кількість операцій над векторами, для практичної реалізації обрано архітектуру SIMD. Цей тип архітектури дає змогу виконати один і той самий потік команд для багатьох потоків даних.

Для процедури побудови дискретних динамічних моделей використано розпаралелення за двома рівнями паралелізму: крупнозернистим та дрібнозернистим [12].

Для дослідження ефективності розпаралелення процесу ідентифікації моделі проведено порівняння часу виконання послідовного алгоритму на центральному процесорі та час виконання з розпаралеленням на графічному процесорі. Досягнута ефективність розпаралелення залежить від кількості точок, для яких обчислюється функція мети на один крок алгоритму оптимізації. Час послідовної програми зі збільшенням згенерованих точок поступово збільшується, а час паралельної програми – залишається відносно сталим [11].

Програма є універсальною для будь-якої відеокарти з підтримкою CUDA. Тому доцільно порівняти виконання паралельної програми на відеоадаптерах з різним показником Compute Capability.

Тестування програми проведено на трьох GPU з різними архітектурами – GeForce GTS 250, GeForce 525M і GeForce GTX 650 Ti. У табл. 1 приведено характеристики трьох відеоадаптерів з різними архітектурами.

Як видно з табл. 1, за Compute Capability графічний процесор має архітектуру Tesla, GeForce 525M – Fermi, GeForce GTX 650 Ti – Kepler.

Порівняння графічних процесорів GeForce GTS 250, GeForce 525M і GeForce GTX 650 Ti

GPU (архітектура)	GeForce GTS 250 (Tesla)	GeForce 525M (Fermi)	GeForce GTX 650 Ti (Kepler)
Кількість транзисторів (млн. шт.)	754	585	2540
CUDA-ядра	128	96	768
Compute Capability	1.1	2.1	3.0
Частота ядра (МГц)	738	600	928
Пропускна смуга пам'яті (ГБ/сек)	70.4	28.8	86.4
Інтерфейс пам'яті	256-bit	128-bit	128-bit GDDR5
Об'єм пам'яті (МБ)	510	1536	1024

Показником ефективності вибрано час виконання паралельної програми, який припадає на одне ядро. На рис. 8 представлено порівняння відношення часу виконання програми до кількості ядер на графічному процесорі. Графік представлено у логарифмічному масштабі.

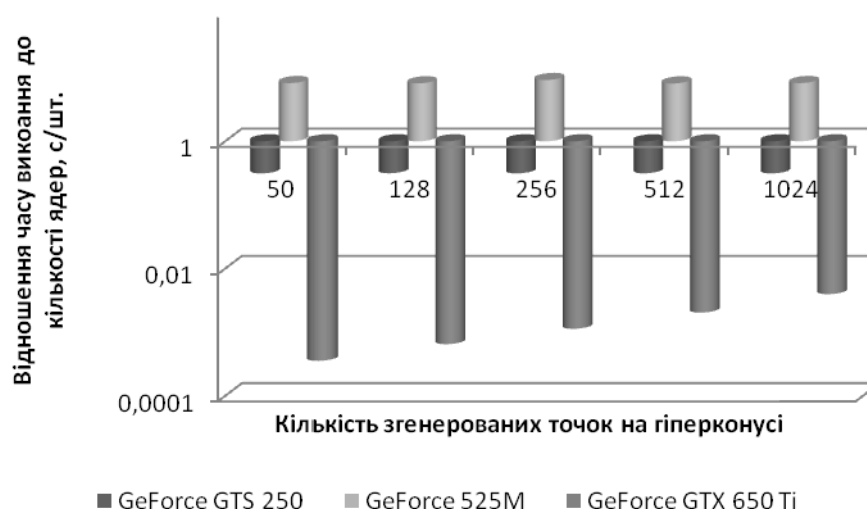


Рис. 8. Порівняння відношення часу виконання програми до кількості ядер на різних графічних процесорах

### Висновки

У цьому дослідженні проведено порівняльний аналіз відеоприскорювачів з різними архітектурами – Tesla, Fermi, Kepler. Проведено чисельні експерименти розпаралелення на GPU з використанням різних архітектур.

Для GPU GeForce 525M показники частоти ядра, пропускної смуги пам'яті, інтерфейсу пам'яті є нижчими, оскільки це відеокарта для мобільних ПК. Тому для цієї відеокарти часова складність виконання програми є найвищою.

Незважаючи на те, що перший відеоадаптер належить до старих архітектур, він має досить хороші характеристики частоти ядра та інтерфейсу пам'яті, що дає змогу виконувати на ньому ефективне розпаралелення.

Незважаючи на велику кількість ядер в новій архітектурі Kepler, ефективність розпаралелення на ній зростає не сильно. Оскільки в цій архітектурі пожертвували продуктивністю заради енергоефективності.

### Література

1. NVIDIA CUDA – доступний билет в мир больших вычислений [Электронный ресурс]. – Режим доступа: <http://www.computerra.ru/interactive/423392/>.
2. NVIDIA Quadro открывает новую эру мобильных суперкомпьютеров [Электронный ресурс]. – Режим доступа: <http://www.nvidia.ru/object/nvidia-quadro-fermi-mobile-20110222-ru.html>.
3. Черняк Л. Многоядерные процессоры и грядущая параллельная революция /Леонид Черняк [Электронный ресурс]. – Режим доступа: <http://www.osp.ru/os/2007/04/4219910/>.
4. Боресков А. В. Параллельные вычисления на GPU. Архитектура и программная модель CUDA/ А. В. Боресков, А. А. Харламов и др.. – М.: Издательство Московского университета, 2012. – 336 с.
5. Стахів П. Г. Метод розпаралелення задачі ідентифікації параметрів дискретних динамічних макромоделей на масивно-паралельних процесорах / П. Г. Стахів, І. П. Струбицька // Наукові нотатки. – Луцьк, 2010. – № 27. – С. 300-305.
6. Струбицька І. П. Оцінка ефективності розпаралелення обчислень на графічних процесорах і

різних операційних системах // Матеріали II Всеукраїнської школи-семінару молодих вчених і студентів «Сучасні комп'ютерні інформаційні технології». – Тернопіль: ТНЕУ, 2012. – С. 143-144.

7. GeForce GTX 680: Чемпион на стероидах [Электронный ресурс]. – Режим доступа: <http://www.3dnews.ru/626473>

8. NVIDIA GeForce GTX 680 Review: Retaking The Performance Crown [Electronic Resource]. – Mode of access: <http://www.anandtech.com/show/5699/nvidia-geforce-gtx-680-review/2>

9. Kepler - the world's fastest, most efficient HPC architecture - [Electronic Resource]. – Mode of access: <http://www.nvidia.com/object/nvidia-kepler.html>

10. Firma NVIDIA wprowadza nowy standard w rozwiązaniach do obliczeń wysokowydajnych – procesory graficzne Tesla oparte na architekturze Kepler - [Elektroniczne zasób]. - Tryb dostępu: <http://www.nvidia.pl/object/hpc-kepler-based-tesla-gpus-20120516-pl.html>

11. Стахів П. Г. Розпаралелення процесу побудови дискретної динамічної моделі двообмоткового трансформатора / Стахів П. Г., Струбицька І. П., Козак Ю. Я. // Вісник Тернопільського національного технічного університету. – 2012. – №1 (65). – С. 182-187.

12. Козак Ю. Я. Розпаралелення алгоритму оптимізації параметрів дискретних динамічних моделей на масивно-паралельних процесорах / Ю. Я. Козак, П. Г. Стахів, І. П. Струбицька // Відбір і обробка інформації. – 2010. – Вип. 32 (108). – С. 126-130.

#### References

1. NVIDIA CUDA – dostupnyi bilet v mir bolshih vichislenii [Elektronnyi resurs]. – Rezhym dostupu: <http://www.computerra.ru/interactive/423392/>.

2. NVIDIA Quadro otkryvaet novuiu eru mobilnyh superkompiuterv [Elektronnyi resurs]. – Rezhym dostupu: <http://www.nvidia.ru/object/nvidia-quadro-fermi-mobile-20110222-ru.html>.

3. Cherniak L. Mnogoiadernnye protsessory i griadushchaia paralelnaia revoliutsiia / Leonid Cherniak [Elektronnyi resurs]. – Rezhym dostupu: <http://www.osp.ru/os/2007/04/4219910/>.

4. Borieskov A. V. Parallelnyye vychysleniia na GPU. Arhitektura i programmnaia model CUDA/ A. V. Boriaskov, A. A. Kharlamov i dr.. – M.: Izdatelstvo Moskovskogo universiteta. – 336 s.

5. Stakhiv P. G. Metod rozparalelennia zadachi identifikatsii parametrv diskretnykh dynamichnykh makromodelei na masyvno-paralelnykh protsesorah / P. G. Stakhiv, I. P. Strubyska // Naukovi notatky. – Luts'k, 2010. – № 27. – S. 300-305.

6. Strubyska I. P. Otsinka efektyvnosti obchyslen na grafichnykh protsesorah i riznykh operatsiinykh systemah // Materialy II Vseukrainskoi shkoly-seminaru molodykh vchenykh i studentiv «Suchasni kompiuterni informatsiini tehnologii ». – Ternopil: TNEU, 2012. – S. 143-144.

7. GeForce GTX 680: Chempion na steroidah [Elektronnyi resurs]. – Rezhym dostupu: <http://www.3dnews.ru/626473>

8. NVIDIA GeForce GTX 680 Review: Retaking The Performance Crown [Electronic Resource]. – Mode of access: <http://www.anandtech.com/show/5699/nvidia-geforce-gtx-680-review/2>

9. Kepler - the world's fastest, most efficient HPC architecture - [Electronic Resource]. – Mode of access: <http://www.nvidia.com/object/nvidia-kepler.html>

10. Firma NVIDIA wprowadza nowy standard w rozwiązaniach do obliczeń wysokowydajnych – procesory graficzne Tesla oparte na architekturze Kepler - [Elektroniczne zasób]. - Tryb dostępu: <http://www.nvidia.pl/object/hpc-kepler-based-tesla-gpus-20120516-pl.html>

11. Stakhiv P. G. Rozparalelennia protsesu pobudovy dyskretnoi dynamichnoi modeli dvoobmotkovogo transformatora / Stakhiv P. G., Strubyska I. P., Kozak Y. Y. // Visnyk Ternopil'skogo natsionalnogo tehnicznego universitetu. – 2012. – №1 (65). – S. 182-187.

12. Kozak Y. Y. Rozparalelennia algorytmu optymizatsii dyskretnykh dynamichnykh modelei na masyvno-paralelnykh protsesorah / Y. Y. Kozak, P. G. Stakhiv, I. P. Strubyska // Vidbir i obrobka informatsii. – 2010. – Vyp. 32 (108). – S. 126-130.

Рецензія/Peer review : 16.11.2014 р.

Надрукована/Printed : 20.10.2015 р.