

А. О. Мельник, Н. Б. Козак

Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин

ПАРАМЕТРИ СИСТЕМИ МАКРОКОМАНД ДЛЯ ГРАФІЧНОГО ПРИСКОРЮВАЧА

© Мельник А. О., Козак Н. Б., 2016

Визначено ключові параметри при генерації системи макрокоманд. Розглянуто залежності рівня прискорення і значень параметрів системи макрокоманд.

Ключові слова: паралельне опрацювання даних, графічні прискорювачі обчислень, генерування програмного коду.

PARAMETERS OF MACROCOMMAND SYSTEM FOR GRAPHICS ACCELERATOR

© Melnyk A. O., Kozak N. B., 2016

The key parameters to generate macrocommand system are defined. The dependencies of acceleration and parameter values for macrocommand system are considered.

Key words: parallel data processing, graphics processing units, computing accelerators, program code generation.

Вступ

Реалізація системи автоматичного генерування коду для графічного прискорювача з коду, написаного мовами високого рівня загального призначення, покликана спростити використання технології GPGPU. Така система має враховувати обмеження, що накладаються архітектурою графічного прискорювача [1]. При виконанні паралельного коду графічним процесором основні обмеження пов'язані з обміном проміжними результатами обчислень через спільну пам'ять, яка розміщена на одному кристалі з графічним процесором. Хоча в графічному прискорювачі передбачено механізм одночасної паралельної роботи кількох процесорних елементів зі спільною пам'яттю, в ньому відсутній механізм паралельного читання кількох елементів даних одним процесорним елементом. Оскільки для обміну даними між процесорними елементами доцільно застосовувати саме спільну пам'ять [2], то варто мінімізувати кількість операцій читання-запису при роботі з нею.

Одним із методів мінімізації кількості звернень до спільної пам'яті може бути композиція фрагментів коду таким чином, щоб частину даних обчислень використовував локально один процесорний елемент. Для зберігання таких локальних даних спільну пам'ять можна буде не використовувати, а залишати їх на регістрах процесорного елемента. Якщо сформувати велику кількість таких фрагментів коду, то вони будуть повторюватись. Подібні фрагменти коду, які відрізнятимуться лише вхідними даними, можна вважати *макрокомандою*. Множина ж таких макрокоманд, достатня для вираження алгоритму, є *системою макрокоманд* при формуванні коду.

Для того самого сирцевого коду можна формувати велику кількість різних систем макрокоманд. Кожну таку систему макрокоманд можна охарактеризувати якимись числовими величинами, що описуватимуть її структуру. Такі величини в роботі названо *параметрами системи макрокоманд*. Деякі параметри системи макрокоманд впливатимуть на рівень прискорення при використанні графічного прискорювача.

1. Постановка задачі

Потрібно визначити вимоги до процесу вибору та параметри системи макрокоманд при формуванні коду для графічного прискорювача, а саме:

- визначити параметри системи макрокоманд, які впливатимуть на рівень прискорення при використанні графічного прискорювача;
- визначити залежність рівня прискорення від значень параметрів системи макрокоманд;
- обчислити мінімальні числові значення параметрів, при яких виконання коду графічним прискорювачем буде давати ефект збільшення продуктивності у порівнянні з центральним процесором.

2. Система макрокоманд для виконання графічним прискорювачем

Вибір системи макрокоманд розглядається в контексті роботи програмної системи, наведеної на рис. 1. Ця система перетворює сирцевий код, поданий мовою високого рівня загального призначення, на код, призначений для виконання комп'ютерною системою на основі центрального та графічного прискорювачів.

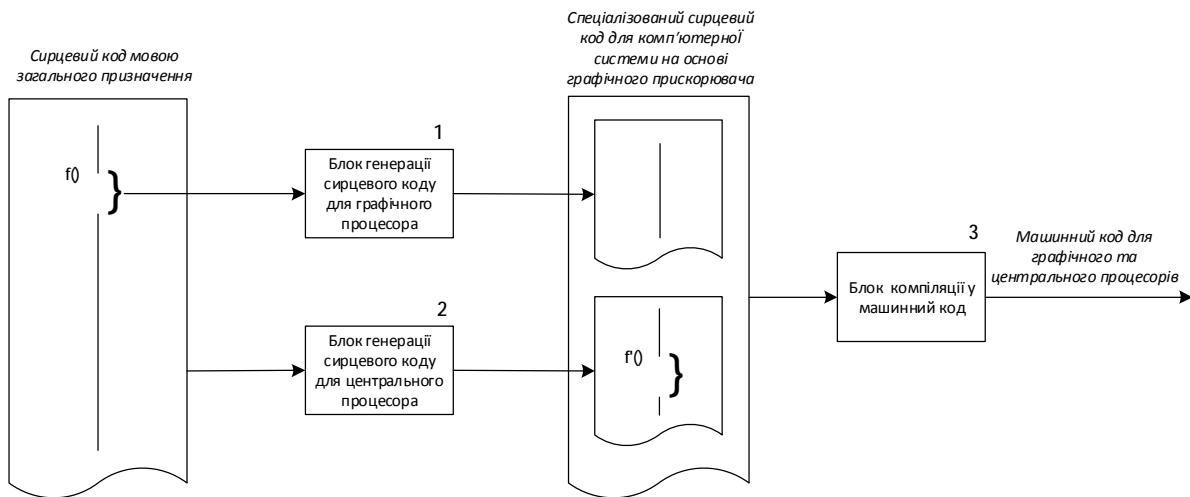


Рис. 1. Загальна схема роботи системи

Блок генерування сирцевого коду для графічного процесора перетворює виокремлену частину коду для виконання графічним прискорювачем (рис. 2). Блок автоматично враховує всі особливості графічного процесора, і згенерований ним код максимально завантажує його функціональні елементи для досягнення високої продуктивності.



Рис. 2. Блок формування сирцевого коду для графічного процесора

Початкова МКМД-модель обчислень, яка формується за допомогою блоків 1.1 та 1.2, призначена для виконання на обчислювачі з архітектурою МКМД без обмежень на обмін даними між процесорними елементами. Така МКМД-модель обчислень може бути ефективно реалізована внаслідок синтезу обчислювача зі спеціалізованою комутуючою мережею на основі ПЛІС [3] і не призначена для виконання графічним процесором. Вона формується лише для виявлення будь-яких можливостей паралельного виконання команд. Такі блоки можна виконати за допомогою програмних засобів LLVM [4].

Для формування кінцевої МКМД-моделі блоком 1.4 у системі макрокоманд ключову роль відіграє робота блоку 1.3, що формує цю систему. Числові значення параметрів системи макрокоманд мають задати набір команд, який буде достатнім для генерації ефективного сирцевого коду, одержаного після компіляції МКМД-моделі блоком 1.5.

3. Параметри системи макрокоманд

Раніше в роботі [5] було показано, що мінімальне прискорення (S') від розпаралелення алгоритму для графічного прискорювача має відповідати такому співвідношенню:

$$S' > \frac{S'''}{S''},$$

де S''' – деяке мінімальне прискорення, при якому використання додаткового обладнання є доцільним; S'' – сповільнення прискорення через механізм обміну результатами обчислень між центральним та графічним процесорами.

Якщо визначити мінімальне прийнятне прискорення $S''' = 1,5$ і знехтувати сповільненням прискорення внаслідок обміну результатами обчислень між центральним та графічним процесорами $S'' \approx 1$ (припускаючи, що початковий час виконання кожного фрагмента коду значно перевищує час вводу-виводу даних), то, враховуючи двократну різницю частот роботи центрального і графічного процесорів, прискорення S' має мати значення, більше за 3.

Крім того, прискорення S' відрізнятиметься від теоретичного прискорення, якого досягають розпаралеленням операцій, і залежатиме від процесу формування макрокоманд.

Нехай макрокоманди формуються з послідовності базових операцій проміжного подання для початкової МКМД-моделі. Тоді такі макрокоманди можна описати двома параметрами: N – кількість послідовних етапів при формуванні макрокоманди та M – кількість команд одного етапу обчислень. Значення M є менш важливим і може дорівнювати 1, оскільки операції одного етапу обчислень не мають жодних залежностей за даними і генератор коду може їх довільно розподілити за процесорними елементами.

При формуванні системи макрокоманд її розмір T має бути мінімальним і наближатися до співвідношення ($T < HL$), де L – кількість багатопотокових блоків, в яких групи потоків виконуються незалежно від інших груп потоків; H – загальна кількість етапів обчислень у системі макрокоманд.

Однотипні макрокоманди, що можуть бути виконані одночасно, мають бути згруповані, а їх кількість P у кожній групі має наближатися до співвідношення ($P \sim K/L$), де K – загальна кількість макрокоманд одного етапу, які можуть бути виконані одночасно.

Враховуючи, що кількість команд у макрокоманді на одну операцію читання-запису фактично визначатиме кількість звернень до спільної пам'яті, можна задати деякий коефіцієнт E для визначення мінімальної такої кількості. Тоді можна сформувати співвідношення ($(N_{in} + N_{out}) < MN / E$), де N_{in} – кількість вхідних даних для однієї макрокоманди; N_{out} – кількість вихідних даних для однієї макрокоманди.

Отже, робота програмного блоку вибору системи макрокоманд задається значеннями N , M , E та L .

4. Розрахунок мінімальних значень M та N для відповідних рівнів прискорення

Для макрокоманди, що була сформована з деякої ділянки початкової МКМД-моделі, крім команд для виконання самої макрокоманди, буде потрібна ще деяка кількість додаткових команд, яку можна оцінити як $(o(c + h)MN + g)$, де o – кількість операцій читання-запису для однієї операції проміжного подання; c – кількість тактів для забезпечення механізму читання-запису одного

елемента до спільної пам'яті; h – кількість тактів на виконання ділянки програмного коду для формування адреси вибірки даних для етапів обчислень; g – кількість тактів на виконання ділянки програмного коду для реалізації вибору макрокоманди. Допускаючи, що при об'єднанні деякого фрагмента початкової МКМД-моделі всі дані внутрішніх етапів обчислень цього фрагмента локалізовані в його області, оцінка кількості додаткових команд набуде вигляду $(o(c + h)M + g)$, оскільки при такому припущенні вхідні дані читаються лише на першому внутрішньому етапі обчислень, а записуються результати обчислень лише на останньому внутрішньому етапі обчислень. Ця додаткова кількість команд виконується паралельно на всіх процесорних елементах, але є тією частиною коду, яка далі не розпаралелюється та завжди присутня в кожній макрокоманді та не залежить від кількості внутрішніх послідовних етапів обчислень, з яких була сформована сама макрокоманда. Тому, враховуючи кількість команд nMN , які розпаралелюються, частку коду, яка не розпаралелюється, можна визначити так:

$$p = \frac{o(c+h)M + g}{o(c+h)M + g + nMN} \quad (1)$$

Використовуючи закон Амдала $S = \frac{1}{p + \frac{1-p}{n}}$, можна отримати таке співвідношення для

знаходження значення N :

$$N = \frac{o(c+h)M + g}{M} \frac{S-1}{n-S} \quad (2)$$

Оскільки більшість команд проміжного подання є триадресними, то $o = 3$. Враховуючи механізм [1] читання проміжних даних, що передбачає два звернення до спільної пам'яті, а також те, що при безконфліктному зверненні до спільної пам'яті витрачається 2 такти, можна оцінити час читання-запису одного елемента даних в 4 такти ($c = 4$). Значення кількості тактів на виконання ділянки програмного коду для формування адреси вибірки даних проміжних етапів обчислень та значення кількості тактів для реалізації вибору макрокоманди відповідно рівні 8 та 10 ($h = 8, g = 10$). Зважаючи на те, що тільки потоки в межах однієї групи з 32-х потоків, так званого варпу (*англ. warp*), виконуються фізично одночасно, робити розрахунок мінімальних значень N і E потрібно для $n = 32$ потоків, оскільки лише в межах цієї групи потоків можна реалізувати дрібнозернистий вид паралелізму, а потоки різних варпів можуть перебувати на різних стадіях виконання програми і не будуть забезпечувати синхронності. Тож отримуємо такий вираз:

$$N = \frac{26M + 10}{M} \frac{S-1}{32-S} \quad (3)$$

Обчислені значення наведено в таблиці, починаючи від $S=3$, оскільки таке значення є мінімально допустимим.

Мінімальні значення N і E для різних значень S і M

Очікуване прискорення		M				
		1	2	3	4	M → ∞
S=3, при n = 32 (~10,67%)	N	3(~2,49)	3(~2,14)	3(~2,02)	2(~1,97)	2(~1,79)
	E	~0,83	~0,71	~0,67	~0,66	~0,6
S=4, при n = 32 (12,5%)	N	4(~3,86)	4(~3,32)	4(~3,14)	4(~3,05)	3(~2,79)
	E	~1,29	~1,11	~1,05	~1,02	~0,93
S=8, при n = 32 (25%)	N	11(10,5)	10(~9,04)	9(~8,56)	9(8,31)	8(~7,58)
	E	3,5	~3,01	~2,85	2,77	~2,53
S=16, при n = 32 (50%)	N	34(33,75)	30(~29,07)	28(27,5)	27(~26,72)	25(~24,38)
	E	11,25	~9,69	~9,17	~8,91	~8,13
S=24, при n = 32 (75%)	N	104(103,5)	87(~86,13)	85(~84,33)	82(~81,94)	75(~74,75)
	E	34,5	~28,71	~28,11	~27,31	~24,75
S=31, при n = 32 (97%)	N	1080	930	880	855	780
	E	360	310	~293,33	285	260

З наведеної таблиці видно, що мінімальне прийнятне значення $N = 3$ або 2 для $M > 3$. Водночас рівень прискорення, що становить 50% від максимального, досягається при використанні більше трьох десятків команд для формування макрокоманди.

Висновки

1. На основі аналізу залежності продуктивності графічного прискорювача від складу макрокоманд визначено ключові параметри системи макрокоманд для графічного прискорювача, а саме:

- кількість послідовних етапів обчислень початкової МКМД-моделі для формування макрокоманди;
- кількість паралельних операцій етапу обчислень початкової МКМД-моделі для формування макрокоманди;
- значення співвідношення суми команд макрокоманди до суми вхідних та вихідних даних цієї макрокоманди;
- кількість багатопотокових блоків, в яких групи потоків виконуються незалежно від інших груп потоків.

2. Визначення ключових параметрів системи макрокоманд дозволило встановити вимоги до роботи блоку вибору системи макрокоманд для формування ефективного коду для графічного процесора.

3. Отримано аналітичні вирази, які встановлюють залежність параметрів системи макрокоманд від рівня прискорення.

4. Отримано мінімальні значення встановлених параметрів для різних рівнів прискорення.

1. Мельник А. Врахування особливостей графічного процесора в процесі створення засобів автоматичного розпаралелення програм / А. Мельник, Н. Козак // Вісник Нац. ун-ту "Львівська політехніка", 2013. – № 751. – С. 3–8. 2. Козак Н. Реалізація паралельних обчислень в графічних прискорювачах [Текст] / Н. Козак // Conference ACSN-2011. – Львів, 2011. – С. 47–49. 3. Мельник А. О., Сало А. М., Клименко В., Цигилик Л., Юрчук А. ХАМЕЛЕОН – система високорівневого синтезу спеціалізованих процесорів // Наук.-техн. журн. Національного аерокосмічного університету ім. М.Є. Жуковського "Харківський авіаційний інститут". – Харків, 2009. – № 5. – С. 189–195. 4. Grosser T., Zheng H., Aloor R., Simbürger A., Großlinger A., Pouchet L.-N. Polly – Polyhedral Optimization in LLVM // International Symposium on Code Generation and Optimization (Charmonix, France, April 2–6, 2011). 5. Мельник А. Підходи до програмування прискорювачів обчислень / А. Мельник, Н. Козак // Вісник Нац. ун-ту "Львівська політехніка". – 2015. – № 830. – С. 68–74.