

4. Орлов С. В. Технологии разработки программного обеспечения // Учебное пособие. 2-е изд. – Санкт-Петербург: Питер, 2003. – 480 с.

5. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации / – Санкт-Петербург: Питер, 2002. – 688 с.
Рукопис подано до редакції 29.03.12

УДК 004.271.32

Ю.В. ЮРКО, канд. техн. наук, О.О. КРЕПЧУК, студент
КПІ «Криворізький національний університет»

НЕГРАФІЧНІ ОБЧИСЛЕННЯ НА ВІДЕОКАРТАХ

Розглянуто відкриту мову обчислень OpenCL та алгоритм використання фреймворку для обчислення неграфічних операцій на графічному процесорі. Розроблено програму, яка демонструє, що паралельне обчислення в матричних операціях (додавання, віднімання і множення на скаляр) суттєво швидше ніж, у класичних методах з використанням центрального процесора.

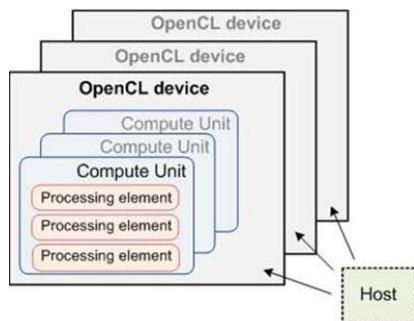
Проблема і зв'язок з практичними задачами. Сьогодні використання більш швидких алгоритмів, пристроїв заощаджують час, зростає попит на нові технології, розробки програмного забезпечення. Зростає обсяг інформації. Мобільність в наш час відіграє велику роль. Тому виникає необхідність застосовувати нові технології програмування, які дозволяють заощаджувати час, використовуючи паралельне обчислювання, наприклад у моделювання геофізичних процесів.

Постановка завдання. Для вирішення цього питання часто застосовується обчислення на GPU. Архітектура графічних процесорів дає змогу використовувати паралельне обчислення (SIMD). Планується реалізувати програму на мові OpenCL в середовищі Microsoft Visual Studio C++ 2008. OpenCL дозволяє використовувати не тільки GPU, а також CPU та DSP.

Аналіз досліджень та публікації. Розв'язком проблеми швидкої та ефективної паралельної обробки надвеликих масивів даних на основі спеціалізованих системних рішень, зокрема паралельно-ієрархічних систем, є використання відеоадаптерів для обчислень загального призначення (GPGPU). Необхідно відзначити, що найбільшій ефективності обчислень на GPU вдається досягти під час виконання матричних обчислень. За теперішніх обсягів пам'яті на GPU можна перемножувати матриці розмірністю вище 10000×10000 , при цьому дані будуть розміщені у відеопам'яті. Така продуктивність досягається за рахунок використання блокових алгоритмів множення матриць, а класичні реалізації на порядок повільніші. За класифікацією по Флінну, графічний процесор відноситься до обчислювальних систем із загальним потоком команд SIMD (Single Instruction Multiple Data - одиничний потік команд і множинний потік даних). Такий процесор називається мультипроцесором. Центральні процесори відносяться до систем з одним потоком команд і одним потоком даних (SISD - Single Instruction Single Data), тобто одночасно виконують одну інструкцію над одним потоком даних. GPU використовують SIMD. Це зумовлено тим, що колір кожного пікселя екрана обчислюється незалежно від інших пікселів по одному і тому ж алгоритму.

Отже, щоб успішно використовувати відеокарту для обчислень, потрібно створити алгоритм з великою кількістю операцій, незалежних одна від одної.

Рис. 1. Схематичне представлення OpenCL-платформи



Для розробки програм під GPU використовують такі технології: OpenCL, Cuda, Ati stream, HLSL, GLSL, Brook+ та інші. Графічні процесори відрізняються архітектурою, обсягом пам'яті, типом пам'яті, частотою ядра та ін. Одним з найбільш відомих виробників GPU є Nvidia, Intel, Ati AMD. Програма на мові OpenCL підтримується відеокартами Ati radeon та Nvidia. OpenCL задіє можливості графічних процесорів для неграфічних обчислень, дозволяючи виконувати їх паралельно.

Модель платформи. Центральним елементом такої моделі виступає поняття хост (host) - первинний пристрій, який управляє OpenCL-обчисленнями, пристроями (CPU, GPU, DSP). Host представлений в єдиному екземплярі.

Модель обчислень. OpenCL складається з хостової програми і набору ядер (kernels). OpenCL-ядро в загальному вигляді являє собою функцію, написану мовою OpenCL C. Ядро

створюється в хостовій програмі і за допомогою спеціальної команди ставиться в чергу на виконання в одному з OpenCL-пристроїв. Одиницею виконання є work item. Work group - набір work item, між якими можливі які-небудь взаємодії. Усі work-item у межах однієї work-group виконуються паралельно на елементах одного обчислювального модуля (OpenCL-device).

Обчислення будуть виконуватися на CPU Athlon x2 250 і GPU Radeon HD 4850 (відео чіп RV770). Основна частина RV770 складається з десяти SIMD ядер, кожне з яких містить по 16 блоків суперскалярних потокових процесорів. Суперскалярність цього відеочіпу така, як і в попередника RV670, тому він містить 800 скалярних 32-бітних потокових процесорів. Для 64-бітових розрахунків подвійної точності використовуються ці ж блоки, знижується тільки темп розрахунків.

SIMD ядра можуть обмінюватися один з одним інформацією за допомогою 16 кілобайтної глобальної пам'яті.

Викладання матеріалу та результати. Основний алгоритм використання мови OpenCL C для операцій над матрицями та векторами.

Виконується перевірка (1), чи підтримує наш пристрій OpenCL, це можливо за допомогою програми GPU-Z. Підготовка масивів до додавання, віднімання та добутку. Вибір розмірності масиву (2). Ініціалізація пристрою `cntx = clCreateContextFromType (NULL, CL_DEVICE_TYPE_ALL, NULL, NULL, NULL)`. Параметр `CL_DEVICE_TYPE_ALL` може використовувати будь-який тип процесора. Оскільки ми використовуємо GPU, то параметр змінюємо на `CL_DEVICE_TYPE_GPU` (3).



Рис. 2. Алгоритм обчислення масивів з використанням OpenCL

У блоці чотири виділяється пам'ять на прискорювачі. Команда `clCreateBuffer` приймає вказівник на вхідний масив. Передачу даних в ядро (host) здійснює ця ж команда `clCreateBuffer` (5). Для запуску обчислень на прискорювачі необхідно написати команду, яка вказуватиме, як саме потрібно використовувати пам'ять при передачі з глобальної (повільної) в більш швидку локальну. Повертаємося в основну програму. Швидкості виконання операцій (додавання, віднімання, множення на скаляр) на GPU та CPU виводяться на екран.

Для використання програми потрібно встановити драйвер відеокарти з підтримкою OpenCL. Для компіляції використовується Microsoft Visual C++ 2008.

```

CPU:
Add - 156 msec
Sub - 140 msec
MulSc - 109 msec
GPU:
Add - 78 msec
Sub - 78 msec
MulSc - 78 msec
    
```

Рис. 3. Результати операцій над матрицями

На рис. 3 показано, що для арифметичних операцій над масивами (розмірності 10000×10000) використовувалися GPU і CPU. Операція *Add* на CPU - це додавання матриці, *Sub* - віднімання, *MulSc* - добуток матриці на скаляр. Нижче такі ж дії, але з використанням GPU.

Висновок та напрям подальших досліджень. У результаті виконаної роботи можна зробити висновок, що за допомогою GPU процесорів з підтримкою OpenCL обробка досить великого обсягу математичних даних є набагато швидшою, ніж на CPU процесорі. Такої швидкості вдалося досягти завдяки паралельній обробці даних, що використовується на графічному процесорі. Тобто при обробці великих масивів даних, отриманих за допомогою геофізичних досліджень, доцільно використовувати технологію GPGPU, оскільки це забезпечує значний вигравш у часі виконання обчислень, що підвищить ефективність роботи.

Список літератури

1. CMsoft. Режим доступу: <http://www.cmsoft.com.br>.
2. Дизайн OpenCL. Режим доступу: <http://opencl.ru/design>.
3. Павловська Т.А. Програмування на мові високого рівня C / C++/Т.А. Павловська, -М.,2002. - 450 с.
4. Ласло М. Обчислювальна геометрія та комп'ютерна графіка на C++/ М. Ласло, - М.: БІНОМ, 1997. - 304с.