Гойчук Д.В., асистент,
Кравченко С.М., старший викладач,
Житомирський державний технологічний університет

АРХІТЕКТУРА І ПРОГРАМУВАННЯ МАСИВНО ПАРАЛЕЛЬНИХ ПРОЦЕСОРІВ

Практично всі сегменти ринку напівпровідникової продукції, включаючи персональні комп'ютери, ігрові консолі, мобільні пристрої, сервери, суперкомп'ютери і мережеві пристрої переходять до використання паралельних платформ. По-перше, паралельні процесори надають більш ефективне використання доступної площі кристала і бюджету енергоспоживання для багатьох вимогливих додатків. По-друге, велика кількість завдань, які традиційно вирішувалися з використанням спеціалізованих інтегральних схем, тепер можуть бути реалізовані на паралельних процесорах, що дозволяє досягти нового рівня функціональності і скоротити витрати на їх розробку. Центральним завданням є розробка додатків, які б найбільш ефективно використовували можливості паралельної архітектури для досягнення поставлених цілей по ефективності і продуктивності.

Технологія CUDA - це програмно-апаратна обчислювальна архітектура Nvidia, яка основана на розширені мови C, яка надає можливість організації доступу до набору інструкцій графічного прискорювача і управління його пам’яттю при організації паралельних обчислень. CUDA допомагає реалізувати алгоритми, які здійснені на графічних процесорах відеоприскорювачів Geforce восьмого покоління і старше (серії Geforce 8, Geforce 9, Geforce 200), а також Quadro і Tesla.

Основні характеристики CUDA:

- уніфіковане програмно-апаратне рішення для паралельних обчислень на відеочіпах Nvidia;
- великий набір підтримуваних рішень, від мобільних до мультичіпових;
- стандартна мова програмування C;
- стандартні бібліотеки чисельного аналізу FFT (швидке перетворення Фур'є) і BLAS (лінійна алгебра);
- оптимізований обмін даними між CPU і GPU;
- взаємодія з графічними API OpenGL і DirectX;
- підтримка 32- і 64-бітових операційних систем: Windows XP, Windows Vista, Linux і MacOS X;
- можливість розробки на низькому рівні.

CUDA включає два API: високого рівня (CUDA Runtime API) і низького (CUDA Driver API), хоча в одній програмі одночасне використання обох неможливо, потрібно використовувати або один або інший. Високорівневий працює «зверху» низькорівневого, всі виклики runtime транслюються в прості інструкції, які оброблюються низькорівневим Driver API. Але навіть «високорівневий» API передбачає знання про пристрій і роботі відеочипів Nvidia, занадто високого рівня абстракції там не має.

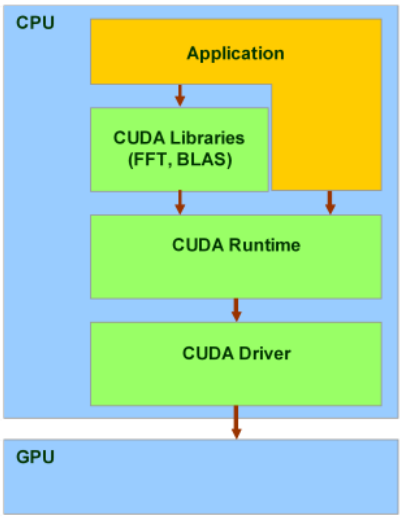

Рис. 1. Склад Nvidia CUDA

Переваги CUDA перед традиційним підходом до GPGPU обчисленням:

- інтерфейс програмування додатків CUDA оснований на стандартнім мові програмування C з розширеннями, що спрощує процес вивчення і впровадження архітектури CUDA;
- CUDA забезпечує доступ до пам’яті, яка розділяється між потоками розміром в 16 Кб на мультипроцесор, яка може бути використана для організації кеша з широкою пропускною здатністю, в порівнянні з текстурними вибірками;
- більш ефективна передача даних між системною і відеопам’яттю;
- лінійна адресація пам’яті, і gather та scatter, можливість запису по довільним адресам;
- апаратна підтримка цілочисельних і бітових операцій.