УДК 621.382

ПРИСКОРЕННЯ АЛГОРИТМІВ ШИФРУВАННЯ РЕАЛІЗОВАНИХ НА ОСНОВІ ГРАФІЧНИХ ПРОЦЕСОРІВ

Пилипчук А.О., Правило В.В.
Інститут телекомунікаційних систем КПІ ім. Ігоря Сікорського, Україна
E-mail: pylypchukangelina@gmail.com

Accelerate encryption algorithms implemented on graphic processors

In modern symmetric algorithms accelerates encryption by implementing them on graphic processors using CUDA and OpenCL technologies.

В сучасних симетричних алгоритмах прискорюють шифрування за допомогою їх реалізації на графічному процесорі з використанням технологій CUDA та OpenCL.

Потужності сучасних відеокарт вистачає тільки для масивно розпаралелених алгоритмів. Для криптографії такі алгоритми можуть викликати падіння криптостійкості через обмеженість у виборі режиму шифрування. З урахуванням останніх тенденцій найкращим рішенням є застосування алгоритмів шифрування на графічних процесорах.

Таким чином, швидке і ефективне використання графічних процесорів для нових алгоритмів шифрування має попит не тільки для збільшення швидкості шифрування, але і для криптоаналізу.

Реалізації розглянутих криптографічних перетворень, призначені для виконання на графічному процесорі, були здійснені за допомогою найбільш популярних на даний момент технологій - CUDA і OpenCL. Вибір технології CUDA обумовлений її перевагою в порівнянні з аналогами в можливостях мови і компілятора, OpenCL має схожі моделі пам'яті і обчислень, що дозволяють здійснювати ефективне використання графічних процесорів, забезпечуючи при цьому кроссплатформенність.

Розглянемо детальніше реалізацію AES шифрування на CUDA GPU.

Advanced Encryption Standard (AES) - симетричний алгоритм блочного шифрування, який шифрує і дешифрує блоки простого тексту і шифрованого тексту за допомогою 128-розрядної, 192-розрядної або 256-розрядного ключа.

CUDA - це програмно-апаратна архітектура паралельних обчислень, що дозволяє істотно збільшити обчислювальну продуктивність завдяки використанню графічних процесорів NVIDIA.

Графічні процесори є відмінною і дуже швидкою лічильною машиною. Але не позбавлені ряду недоліків. Перший з них - вузька сфера застосування. GPU зробили крок далеко вперед центрального процесора в плані нарощування обчислювальної потужності і загальної кількості ядер (відеокарти несуть на собі обчислювальний блок, що складається з більш ніж сотні ядер), проте така висока щільність досягається за рахунок максимального спрощення дизайну самого чіпу.

Ступінь деталізації показує розмір завдання, призначеної для відправки на скалярний процесор. Ступінь деталізації визначає, як розпаралелить алгоритм AES. Важлива стратегія розподілу пам'яті CUDA, тому що є декілька різних типів пам'яті. Особливості кожного типу пам'яті відрізняються в досить сильному ступені.

Визначено ступені деталізації чотирьох типів:

1. Використання розпаралеленого підходу 16 байт / потік означає, що кожний потік незалежно обробляє кожен блок відкритого тексту з 16 байтів. Перевагою реалізації є зниження накладних витрат. Вона не вимагає ніякої синхронізації і загальних даних між потоками. Цей ступінь деталізації використовує паралелізм тільки між блоками відкритого тексту.
2. Ступінь деталізації в 8 байт / потік обробляє один блок відкритого тексту двома потоками. Цей підхід використовує паралелізм між блоками відкритого тексту і внутрішньої обробки відкритого тексту одночасно.
3. Ступінь деталізації в 4 байта / потік обробляє один блок відкритого тексту чотирма потоками. Цей метод відрізняється від 8 байт / потік кількістю потоків для обробки блоку відкритого тексту. Колективна пам'ять і синхронізація необхідна з тієї ж самої причини, як у випадку 8 байт / потік. Ці ступені деталізації використовують більше паралелізму, ніж 16 байт / потік, хоча вони використовують синхронізацію і пам'ять, що розділяється.
4. Краще здійснювати AES обробку, шифруючи, хоча б, 32 бітовими блоками, тому що алгоритм шифрування AES, оптимізований для 32-бітної обробки. Однак він в змозі обробити 1 байт даних в потоках. 1 байт / потік означає, що 16 потоків обробляють блок відкритого тексту узгодженим способом.

Розглянемо доступ до пам'яті при розпаралеленому шифруванні AES на CUDA GPU. AES містить структури даних трьох видів:

1. Звичайний текст, зашифрований текст і проміжні дані.
2. T-box.
3. Раунд ключ (раунд ключ обчислений за секретного ключа).

Коли шифрування розпочато, всі дані зберігаються в пам'яті на хості. На початку обробки AES з використанням CUDA GPU, відкритий текст, раунд ключ і таблиці T-box передаються в глобальну пам'ять і константну пам'ять на GPU. Щоб прискорити обробку, дані передані іншій швидкодійній пам'яті, такий як регістри GPU і колективна пам'ять. Необхідно розглянути особливості цих типів пам'яті для зберігання даних в придатному місці для цих даних. Схема взаємодії центрального і графічного процесора представлена на Рис1.

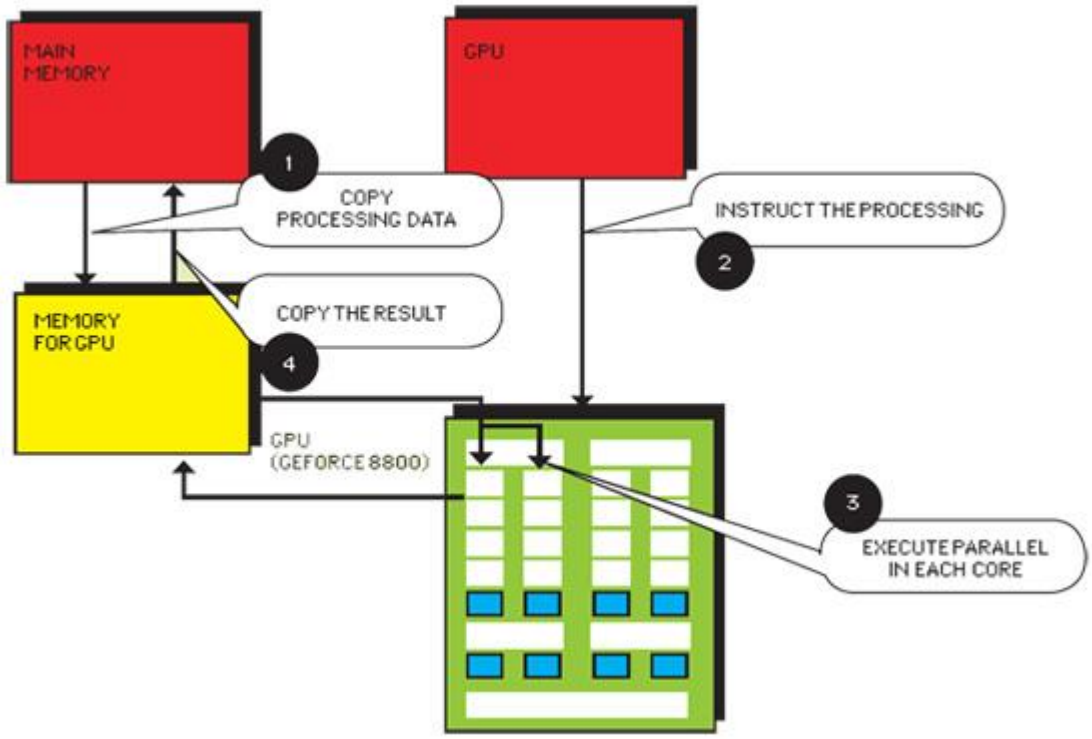

Рис.1. Схема взаємодії між CPU і GPU.

Також необхідно розглянути витрати, які пов'язані з передачею даних між центральним процесором і GPU, щоб ефективно використовувати продуктивність. Для того, щоб приховати ці витрати, CUDA забезпечує перекривання передачі даних та обробки.

На сьогоднішній день існує багато реалізацій, які використовують дані технології. Але ефективність AES шифрування відрізняється за різних умов, таких як ступінь паралельності обробки і різні варіанти використання пам'яті. Тому необхідно визначити за яких умов буде найкращий показник швидкодії.

Література

1. Удальцов В.А., Кармановский Н.С. «Исследование способов скоростной реализации элементов симметричных алгоритмов шифрования при проведении вычислений на графическом процессоре» Журнал Научно-технический вестник информационных технологий, механики и оптики (https://cyberleninka.ru/article/n/issledovanie-sposobov-skorostnoy-realizatsii-elementov-simmetrichnyh-algoritmov-shifrovaniya-pri-provedenii-vychisleniy-na).
2. Гибадуллин Р.Ф., Яковлев А.С., Новиков А.А., Перухин М.Ю. «Ускорение aes шифрования на аппаратно-программной платформе Nvidia CUDA», Журнал Вестник Казанского технологического университета (https://cyberleninka.ru/article/n/uskorenie-aes-shifrovaniya-na-apparatno-programmnoy-platforme-nvidia-cuda).
3. Лебеденко Е.В., "Бондарева Н.В. Возможности технологии CUDA для распараллеливания вычислений в криптографических методах", Журнал Новые информационные технологии в автоматизированных системах (https://cyberleninka.ru/article/n/vozmozhnosti-tehnologii-cuda-dlya-rasparallelivaniya-vychisleniy-v-kriptograficheskih-metodah).
4. Знакомство с программно-аппаратной архитектурой CUDA , O.Montgomeri, 07.03.2018 (https://proglib.io/p/cuda/).