Assignment 3.

Тема: Архитектура GPU и оптимизация CUDA-программ

Задание 1 (25 баллов)

Реализуйте программу на CUDA для поэлементной обработки массива (например, умножение каждого элемента на число). Реализуйте две версии программы:

1. с использованием только глобальной памяти;
2. с использованием разделяемой памяти.
   Сравните время выполнения обеих реализаций для массива размером 1 000 000 элементов.

Задание 2 (25 баллов)

Реализуйте CUDA-программу для поэлементного сложения двух массивов. Исследуйте влияние размера блока потоков на производительность программы. Проведите замеры времени для как минимум трёх различных размеров блока.

Задание 3 (25 баллов)

Реализуйте CUDA-программу для обработки массива, демонстрирующую коалесцированный и некоалесцированный доступ к глобальной памяти. Сравните время выполнения обеих реализаций для массива размером 1 000 000 элементов.

Задание 4 (25 баллов)

Для одной из реализованных в предыдущих заданиях CUDA-программ подберите оптимальные параметры конфигурации сетки и блоков потоков. Сравните производительность неоптимальной и оптимизированной конфигураций.

Контрольные вопросы к Assignment 3

1. Какие основные типы памяти существуют в архитектуре CUDA и чем они отличаются по скорости доступа?
2. В каких случаях использование разделяемой памяти позволяет ускорить выполнение CUDA-программы?
3. Как шаблон доступа к глобальной памяти влияет на производительность GPU-программы?
4. Почему одинаковый алгоритм на GPU может показывать разное время выполнения при разных способах обращения к памяти?
5. Как размер блока потоков влияет на производительность CUDA-ядра?
6. Что такое варп и почему важно учитывать его при разработке CUDA-программ?
7. Какие факторы необходимо учитывать при выборе конфигурации сетки и блоков потоков?

8. Почему оптимизация CUDA-программы часто начинается с анализа работы с памятью, а не с изменения алгоритма?