HPC_matrix_multi_GPU

Лабораторная работа №0

Перемножение матриц.

Задача: реализовать алгоритм перемножения матриц.
Входные данные: 2 матрицы размером от 100x100 до 2000x2000 каждая.
Выходные данные: проверка корректности перемножения + время вычисления
Реализация должна содержать 2 функции перемножения матриц: на CPU и на GPU с применением CUDA.

Для реализации данной задачи использовалась следующая аппаратная база:
Центральный процессор: Intel I5-11300H @ 3,10 GHz.
Оперативная память: Samsung DDR4, 8 GB, 3200 MHz, DualChannel.
Графический процессор: RTX 3060, 6GB VRAM GDDR6.

Функция **matrixMultCPU** как понятно из названия, он осуществляет матричные вычисления в памяти центрального процессора. Для проверки правильности решения используется функция **checkMult**. Её задача состоит в сравнении решений двух алгоритмов.

В функции **matrixMult** 1 элемент итоговой матрицы вычисляется на одной нити. Все индексы, необходимые для вычислений, определяются по индексам блока и нити. У этого подхода имеется определённый недостаток. Элементы крупных матриц должны загружаться несколько раз, находясь при этом в глобальной памяти, что заставляет алгоритм т ратит время на обращение к ней.

Размерность матрицы и размер блоков вводятся вручную в matmul_Korshikov-VI.cu Предположим, что все элементы массива имеют целочисленное значение. Так как массив обладает огромной размерностью(от100 до 2000), Вбивание значений вручную займёт целую вечность, поэтому мы введём их автоматически с помощью **generateRandMatrix**.
Чтобы посмотреть результаты умножения, используем функцию **printMatrix**, она напечатает нам значения в консоль.

Вычисления проводились при размере блока: 16
Результаты вычислений в таблице:

Время выполения и ускорение:

Размерность матрицы	**128**	**512**	**1024**	**2048**
Время вычисления на GPU(распаралеленно), мс.	0,162	4,668	39,177	438,937
Время вычисления на CPU (последовательно), мс.	7,964	861,001	8639,516	169054,770
Ускорение, раз	47,159	184,433	220,524	385,146

Итог

Напрашивается очевидный вывод, что, вычисления на графическом процессоре проходят во много раз быстрее, чем на центральном. Данный эффект можно объяснить количеством этих самых ядер(rtx 3060 имеет 3072 активных CUDA-ядра), что позволяет распределять нагрузку между ними в результате распараллеливания. В то же время, несмотря на большую вычислительную мощность ядер центрального процессора, их во много раз меньше(I5-11300H имеет 4 ядра 8 потоков), из-за чего приходится выполнять все задачи последовательно. При увеличении объёмов задачи отставание только растёт. Всё это указывает на то, что, графический процессор в решении определённых вычислительных задач может оказаться эффективнее центрального.