Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
имени академика С.П. Королева»

ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ

Отчёт по лабораторной работе №4

РЕАЛИЗАЦИЯ ПЕРЕМНОЖЕНИЯ МАТРИЦ ПО ТЕХНОЛОГИИ CUDA

Томашайтис Павел, Барышников Владислав, Елагин Денис

Группа 6313-100503D

Самара 2024

1 Цель работы

1) Реализовать программу перемножения двух матриц с использованием технологии CUDA.

2) Проверить корректность перемножения матриц с использованием технологии CUDA.

3) Измерить статистические характеристики для времени перемножения двух матриц.

2 Реализация программы перемножения двух матриц с использованием технологии CUDA.

Метод перемножения 2 матриц по технологии CUDA представлен на рисунке 1.

```
double cuda_dot(SquareMatrix& rhs)
{
    SquareMatrix result = SquareMatrix(_size);
    int* A = _data;
    int* B = rhs._data;
    int* C = result._data;

    int* Adev = NULL;
    int* Bdev = NULL;
    int* Cdev = NULL;

    cudaMalloc((void**)&Adev, _size * _size * sizeof(int));
    cudaMalloc((void**)&Bdev, _size * _size * sizeof(int));
    cudaMalloc((void**)&Cdev, _size * _size * sizeof(int));

    dim3 threads(BLOCK_SIZE, BLOCK_SIZE);
    dim3 blocks(_size / threads.x, _size / threads.y);

    cudaEvent_t begin, end;
    double time = 0;
    cudaEventCreate(&begin);
    cudaEventCreate(&end);

    cudaEventRecord(begin, 0);
    cudaMemcpy(A, Adev, _size * _size * sizeof(int), cudaMemcpyHostToDevice);
    cudaMemcpy(B, Bdev, _size * _size * sizeof(int), cudaMemcpyHostToDevice);
    matrix_dot << <blocks, threads >> > (Adev, Bdev, Cdev, _size);
    cudaMemcpy(C, Cdev, _size * _size * sizeof(int), cudaMemcpyDeviceToHost);
    cudaEventRecord(end, 0);
    cudaEventSynchronize(end);
    cudaEventElapsedTime(&time, begin, end);

    cudaEventDestroy(begin);
    cudaEventDestroy(end);
    cudaFree(Adev);
    cudaFree(Bdev);
    cudaFree(Cdev);
    return time;
    }
```

Рисунок 1 – Метод перемножения 2 матриц по технологии CUDA.

В данном алгоритме реализована формула: $C_{ij} = \sum_{k=1}^{N} A_{ik}B_{kj}$, суть которой заключается в том, что каждый поток работает со своим блоком матрицы. Программа для перемножения матриц по технологии CUDA представлена в файле kernel.cu.

3 Программа для измерения статистических характеристик, связанных со временем перемножения двух матриц, на языке Python

Программа, написанная на языке Python и представленная в файле statistics.py, позволяет провести статистический анализ по выборке из временных интервалов, полученных при перемножении матриц. Для данной выборки в программе вычисляется среднее, медиана, дисперсия, среднеквадратическое отклонение, коэффициент эксцесса, коэффициент асимметрии, доверительный интервал (надёжность = 0,95).

```
Block size: 16
Matrices 960x960:
Mean: 0.013761899999999999
Median: 0.01370915
Dispersion: 3.483033399999987e-08
STD: 0.00018662886700615173
Skewness: 1.1136044839230623
Kurtosis: -0.1195683987648608
Confidence interval for GAMMA = 0.95: (0.013621172057242599, 0.013902627942757399)
```

Рисунок 2 – Пример работы программы измерения статистических характеристик для матриц 960 на 960 и размера блока 16.

Кроме того, программа в конце своей работы формирует графики зависимости средней величины времени, необходимой для перемножения матриц, от их размера и от размера блока.

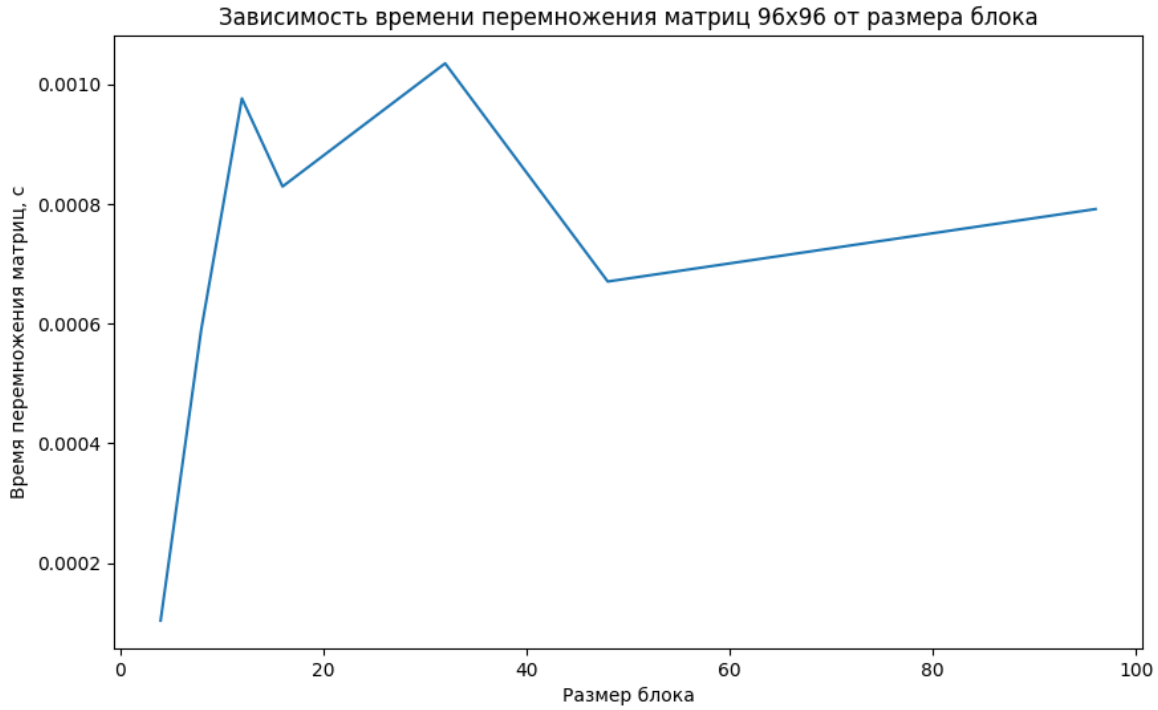

Рисунок 3 – График зависимости времени перемножения матриц размера 96 на 96 от размера блока.

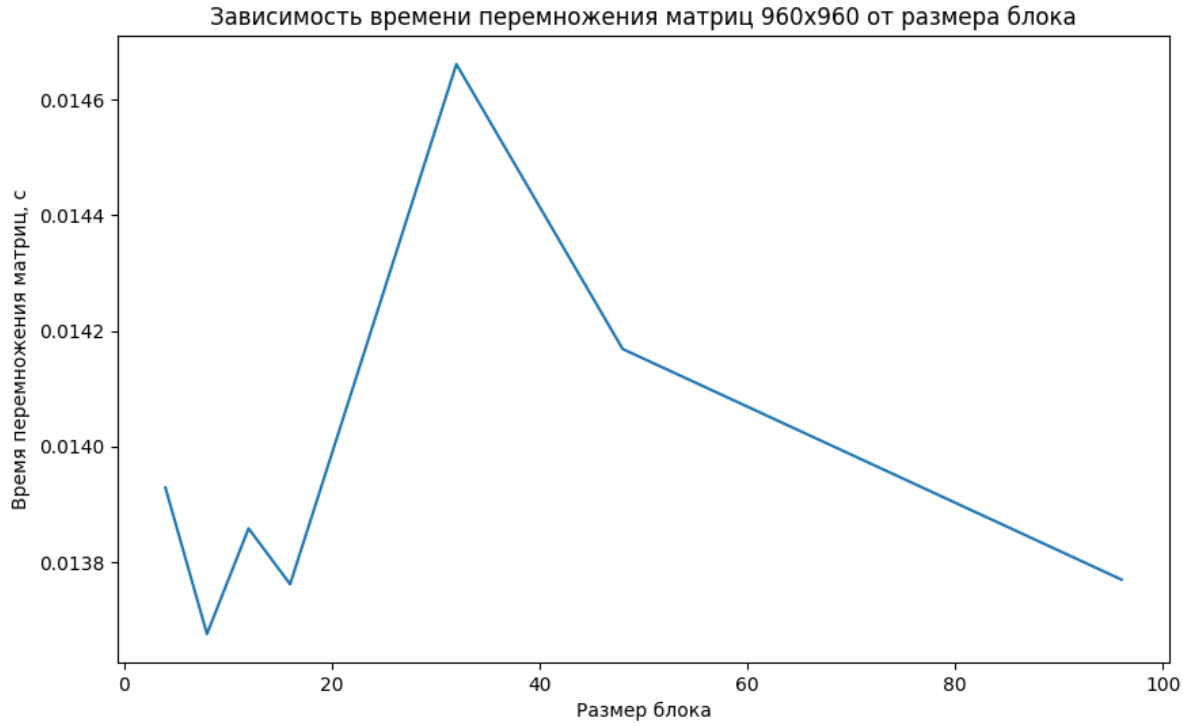

Рисунок 4 – График зависимости времени перемножения матриц размера 960 на 960 от размера блока.

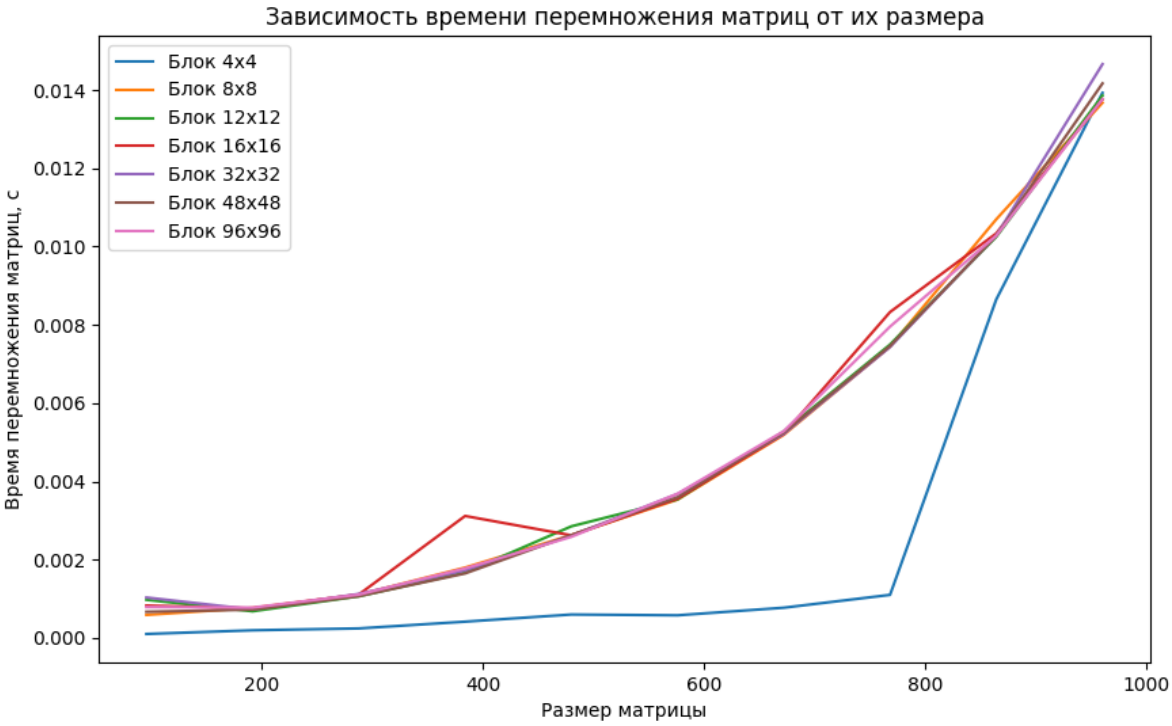

Рисунок 5 – График зависимости времени перемножения матриц от размера.

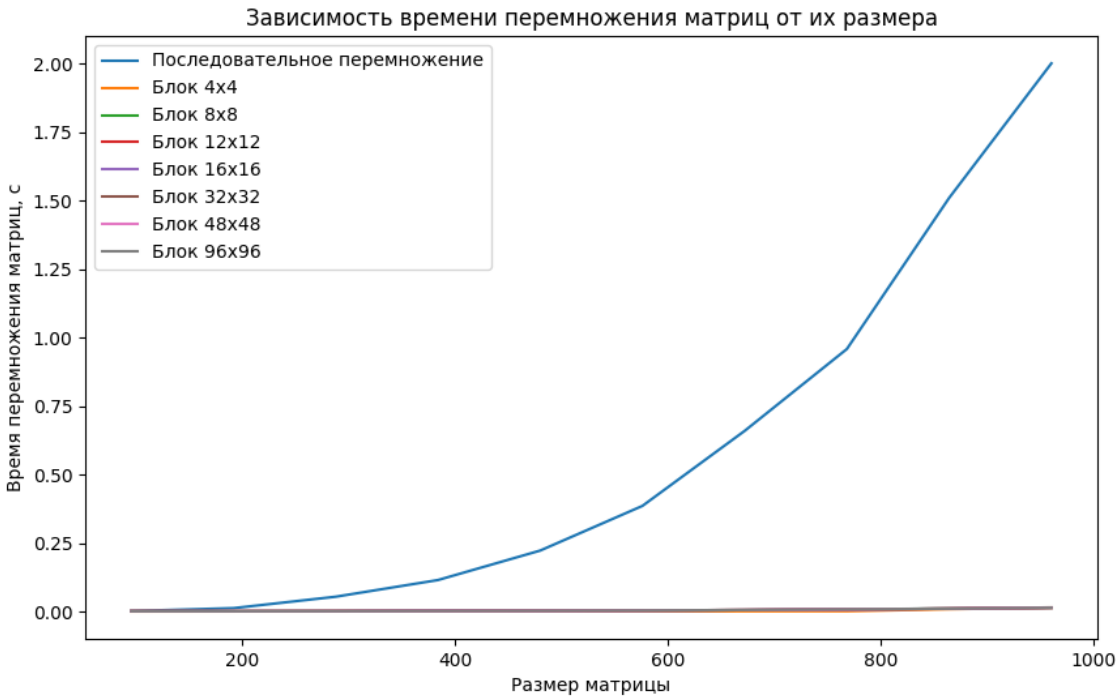

Рисунок 6 – График зависимости времени перемножения матриц от размера.

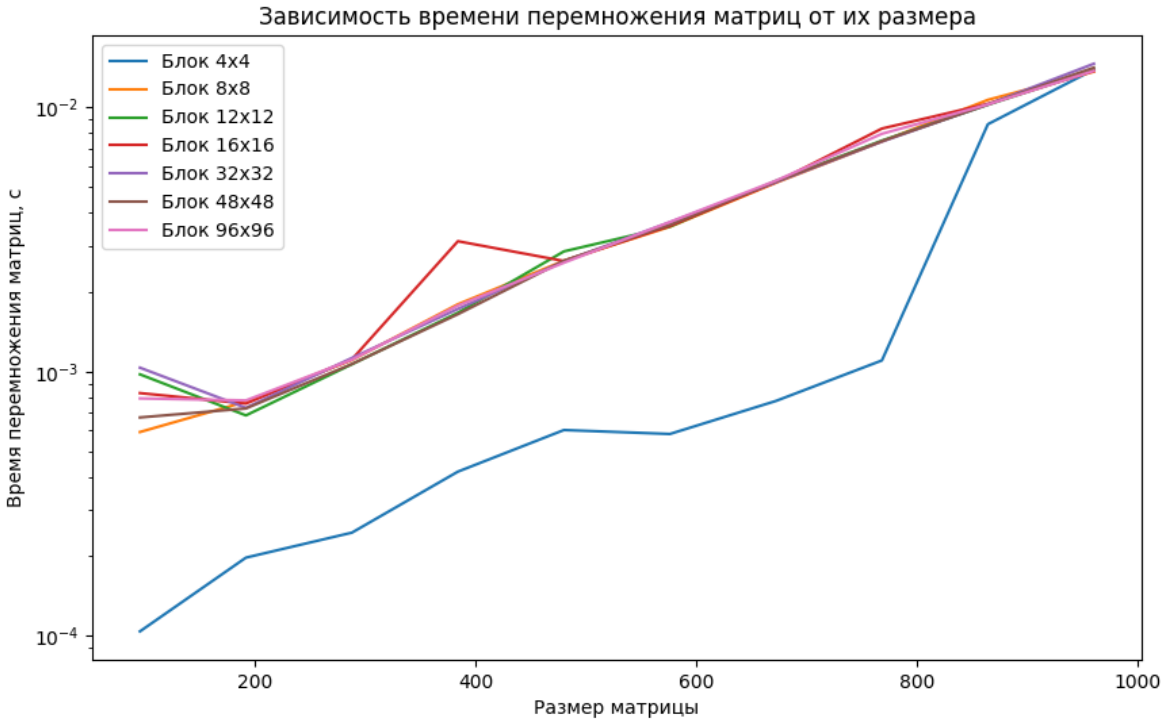

Рисунок 7 – График зависимости времени перемножения матриц от размера.

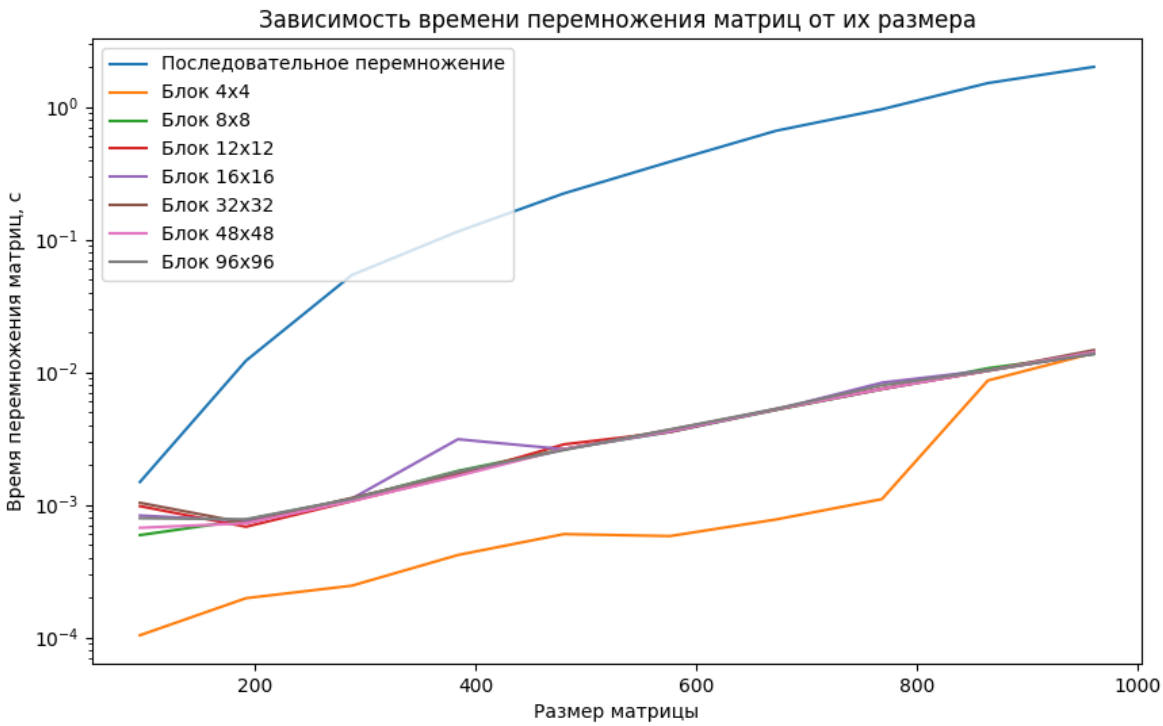

Рисунок 8 – График зависимости времени перемножения матриц от размера.

4 Выводы

В данной лабораторной работе было реализовано перемножение матриц с использованием технологии CUDA и проведено несколько экспериментов по перемножению матриц различного размера при различных размерах блока.

По результатам проведённой работы можно утверждать, что параллельное перемножение двух матриц тем эффективнее, чем меньше блок, но зависит и от размера матрицы. Но по графикам можно сделать однозначный вывод о том, что использование технологии CUDA даёт прирост производительности при перемножении матриц как минимум на порядок.