

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное образовательное

учреждение высшего образования

ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Институт математики, механики и компьютерных наук

имени И. И. Воровича

Направление подготовки:

02.04.02 – Фундаментальная информатика

и информационные технологии

ОТЧЕТ ПО ТЕМЕ:

«Исследование блочного умножения треугольных матриц»

Выполнил:

магистр 1 года 5 группы

Мальцев А.К.

Ростов-на-Дону

2018

Содержание

Постановка задачи (Задание 10)	3
Архитектура компьютера	4
Параметры запусков.....	5
Результаты для компилятора MSVC++	6
Графики работы MSVC++	8
Результаты для компилятора G++	9
Графики работы G++	11
Сравнительный анализ	Ошибка! Закладка не определена.
Заключение	12

Постановка задачи (Задание 10)

Цель данной работы состоит в разработке программы для блочного перемножения матриц, а также в сравнении времени выполнения в зависимости от размера блока.

Согласно условиям задачи, матрицы хранятся в линеаризованном виде. Матрица A верхне-треугольная. Хранится в виде одномерного массива по блочным строкам. Матрица B нижне-треугольная. Хранится в виде одномерного массива по блочным столбцам.

В процессе выполнения работы были реализованы следующие способы выполнения программы:

- 1) Последовательное выполнение программы
- 2) Перемножение каждых двух блоков параллельно
- 3) Параллельное перемножение разных пар блоков в разных вычислительных ядрах одновременно.

Параллельные варианты программы были реализованы при помощи технологии OpenMP.

Проверка корректности перемножения матриц осуществляется путем сравнения результатов вычислений реализованного метода и стандартного перемножения матриц.

Архитектура компьютера

В ходе выполнения экспериментов использовался компьютер со следующими характеристиками:

Процессор: Intel® Core™ i7-8700K

Оперативная память: DDR4, 3000 MHz, 2x8 GB

Технические характеристики процессора:

Семейство - Coffee Lake

Количество ядер - 6

Количество потоков - 12

Базовая тактовая частота 3.7 GHz – 4.7 GHz в режиме Turbo

Частота системной шины 8 GT/s DMI

Кэш-память L1 - 192 КБ

Кэш-память L2 - 1,5 МБ

Кэш-память L3 - 12 МБ

На компьютере установлена операционная система Windows 10.

Параметры запусков

1. Размер матрицы: $N=2880$.
2. Размеры блоков: 1, 6, 10, 15, 20, 24, 30, 36, 40, 60, 72, 80, 96, 120, 144, 160, 180, 240, 360, 480, 720.
3. Типы данных double и float.
4. Параллельные вычисления производились с использованием технологии OpenMP.
5. Компиляторы: Microsoft Visual C++ и g++.
6. Рассчитывалось среднее время за 5 запусков программы.

Результаты для компилятора MSVC++

Размер блока	Время метода 1,с	Время метода 2,с	Время метода 3,с
6	39.1295	131.36	7.06962
10	33.7461	38.2997	6.40732
15	32.3391	18.3366	6.12471
20	33.7273	13.2096	6.28556
24	34.7426	11.5413	6.17309
30	34.4093	10.3671	6.17817
36	34.5187	9.7061	6.37333
40	34.4835	9.48844	6.1519
60	35.271	9.01501	6.28074
72	35.8176	8.8936	6.67714
80	37.0858	9.10948	6.42112
96	44.5976	10.572	7.06909
120	38.6223	9.58128	6.89113
144	42.9454	10.3525	7.84638
160	43.476	10.8739	8.21046
180	49.2959	11.6836	8.7209
240	60.3205	14.2236	9.41252
360	93.1588	23.047	15.6057
480	114.076	22.1843	17.9539
720	142.079	21.1502	21.7044

Компилятор MSVC++ тип данных float. (1 - последовательное выполнение программы;

2 - перемножение каждых двух блоков параллельно; 3 - параллельное перемножение разных пар блоков в разных)

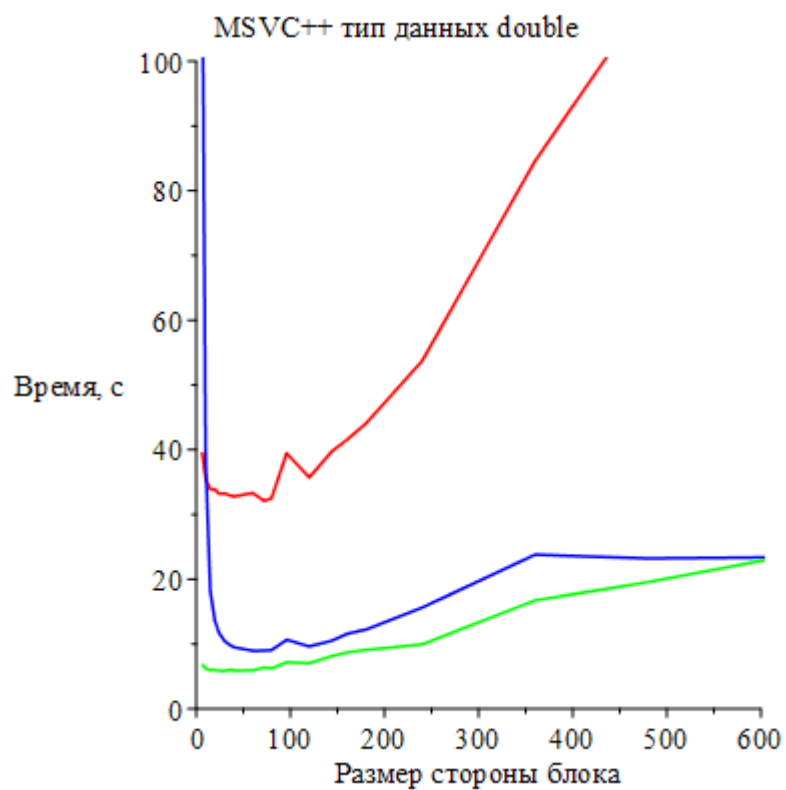
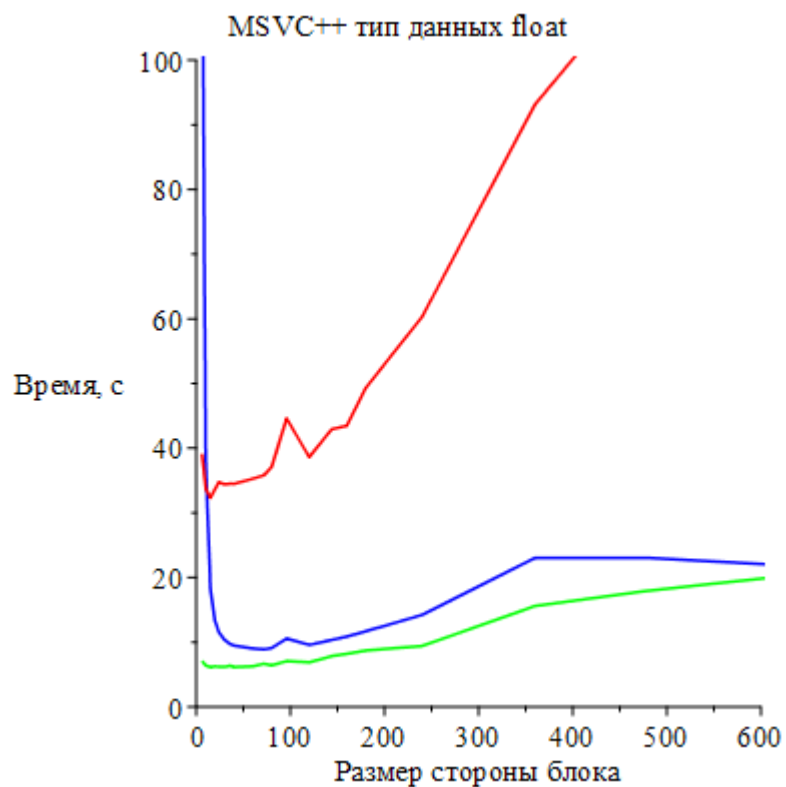
Размер блока	Время метода 1,с	Время метода 2,с	Время метода 3,с
6	39.6496	127.164	6.90885
10	35.4312	37.9161	6.19982
15	33.8808	18.1374	5.98022
20	33.8769	13.4758	5.9141
24	33.2221	11.7491	5.89175
30	33.251	10.4504	5.84571
36	32.9283	9.80458	6.036
40	32.7882	9.49824	5.89563
60	33.3221	8.93581	5.93786
72	32.1026	8.96026	6.36949
80	32.5112	9.0634	6.21702
96	39.4259	10.6565	7.17779
120	35.7093	9.62673	7.04406
144	39.7459	10.513	8.13053
160	41.5337	11.5701	8.68899
180	44.0103	12.2191	9.09298
240	53.706	15.6154	9.93234
360	84.5995	23.8003	16.7359
480	109.902	23.2156	19.5631
720	120.745	23.4888	26.1384

Компилятор MSVC++ тип данных double. (1 - последовательное выполнение программы;

2 - перемножение каждых двух блоков параллельно; 3 - параллельное перемножение разных пар блоков в разных)

Графики работы MSVC++

Красный – метод 1, синий – метод 2, зелёный – метод 3.



Результаты для компилятора G++

Размер блока	Время метода 1,с	Время метода 2,с	Время метода 3,с
6	48.5999		7.88251
10	45.8285	304.739	7.27572
15	44.1004	96.2894	6.71593
20	44.0864	45.4823	6.83885
24	43.9854	28.5316	7.02101
30	43.7955	17.6611	6.81164
36	41.43	13.5595	7.06856
40	44.3438	11.9191	7.13433
60	41.1733	9.24427	7.03406
72	44.7733	8.77552	7.51792
80	46.0708	9.2702	7.44165
96	55.4883	10.4919	8.37469
120	49.5888	9.99127	7.88419
144	50.087	10.3104	8.98318
160	52.8659	10.6425	9.60064
180	61.1548	11.3676	10.3267
240	74.1266	13.9417	11.2786
360	108.177	19.3941	19.5936
480	141.941	24.7189	23.5697
720	166.342	24.9632	27.4628

Компилятор G++ тип данных float. (1 - последовательное выполнение программы;
2 - перемножение каждых двух блоков параллельно; 3 - параллельное перемножение
разных пар блоков в разных)

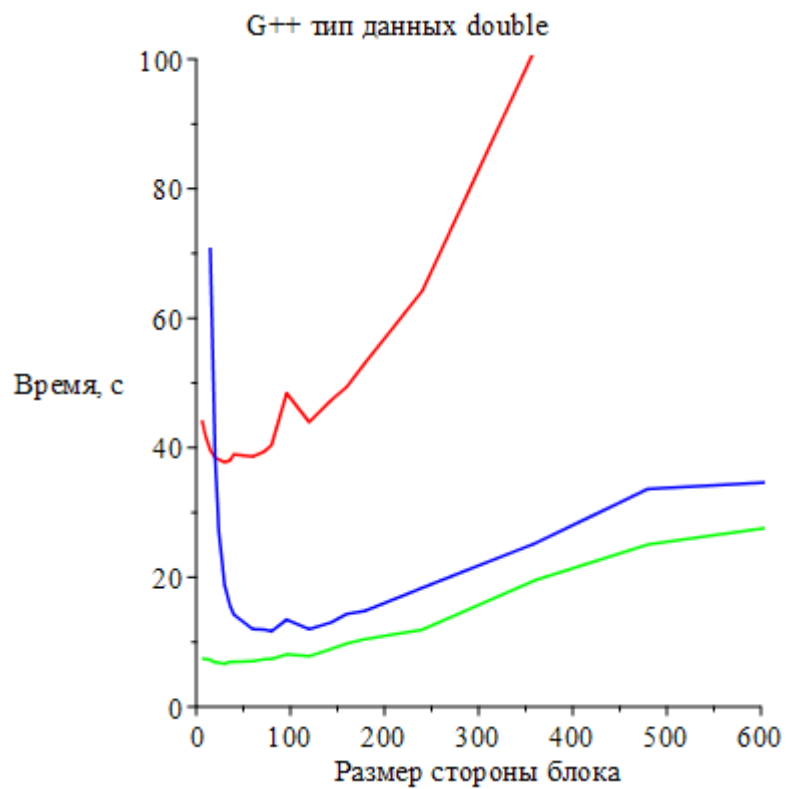
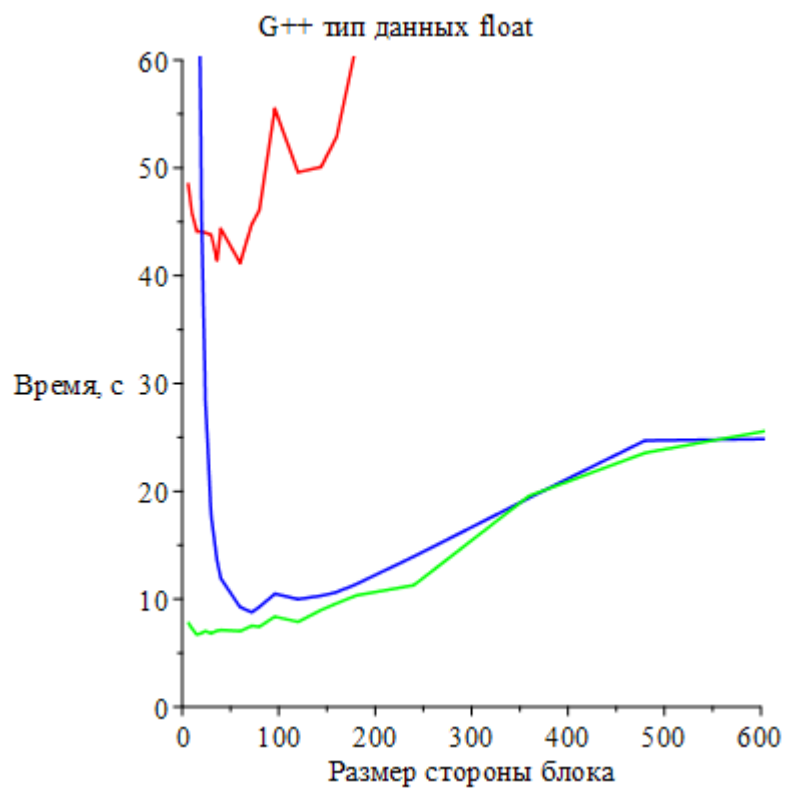
Размер блока	Время метода 1,с	Время метода 2,с	Время метода 3,с
6	44.2546		7.45007
10	41.7812		7.35133
15	39.5931	70.8963	7.26057
20	38.5149	38.7124	6.83571
24	38.1719	26.9958	6.83072
30	37.7636	18.8157	6.6153
36	38.0509	15.5115	6.93448
40	38.9668	14.227	6.94641
60	38.6306	11.9869	7.03322
72	39.3876	11.9381	7.33338
80	40.4149	11.6379	7.40023
96	48.4085	13.463	8.07636
120	43.9691	11.9879	7.80312
144	47.4371	13.07	8.96102
160	49.3912	14.3386	9.77182
180	53.2217	14.8333	10.464
240	64.1788	18.3782	11.8912
360	101.567	25.2325	19.5417
480	137.853	33.644	25.058
720	151.866	35.5769	29.9718

Компилятор G++ тип данных double. (1 - последовательное выполнение программы;

2 - перемножение каждых двух блоков параллельно; 3 - параллельное перемножение разных пар блоков в разных)

Графики работы G++

Красный – метод 1, синий – метод 2, зелёный – метод 3.



Анализ результатов

По результатам проведенных экспериментов видно, что матрицы типа float перемножаются быстрее, чем матрицы типа double. Это обусловлено тем, что элементов типа float (4 байта) в кэш попадает больше, чем элементов типа double (8 байт). Отсюда сокращение количества обращений к памяти, что снижает время работы программы.

Программа, скомпилированная в G++ в среднем работает дольше той, которая скомпилирована в MSVC++.

Также можно заметить, что метод 3 работает быстрее метода 2.

Ускорение по сравнению с последовательным перемножением блоков определенного размера примерно равно количеству ядер процессора.

Заключение

В результате работы была реализована программа умножения верхне-треугольной матрицы на нижне-треугольную, было произведено распараллеливание двумя способами. Проведены исследование и сравнительный анализ временных показателей трёх реализованных методов, двух разных компиляторов и двух типов данных.

Параллельные варианты реализации ускоряют работу программы. При малых размерах блоков оптимален метод распараллеливания двух блоков, а при больших размерах – перемножение разных пар блоков в разных вычислительных ядрах.