МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное

учреждение высшего профессионального образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт математики, механики и компьютерных наук им. И.И.Воровича

Кафедра алгебры и дискретной математики

Отчет

Тема: Блочные вычисления. Модели времени выполнения программ. Блочные размещения массивов, дополняющие блочные вычисления.

**Выполнила:**

студентка 4 курса 1 группы

Лелюк Анастасия Андреевна

**Преподаватель:**

Баглий Антон Павлович

Ростов – на – Дону – 2018

**Содержание**

1. Постановка задачи
2. Алгоритм решения
3. Проверка работы программы
4. Полученные результаты
5. Характеристики компьютера
6. Вывод

**Постановка задачи**

Вариант 15  
 Написать программу блочного умножения двух матриц C = A\*B.   
Матрица A верхне-треугольная. Хранится в виде одномерного массива по блочным столбцам. Матрица B верхне-треугольная. Хранится в виде одномерного массива по блочным строкам.

Распараллелить блочную программу умножения двух матриц C = A\*B с использованием технологии OpenMP. Определить оптимальные размеры блоков в обоих случаях.

Определить оптимальные размеры блоков в обоих случаях. Провести численные эксперименты и построить таблицу сравнений времени выполнения различных программных реализаций решения задачи. Определить лучшие реализации. Проверить корректность (правильность) программ.

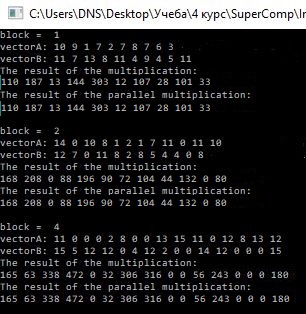
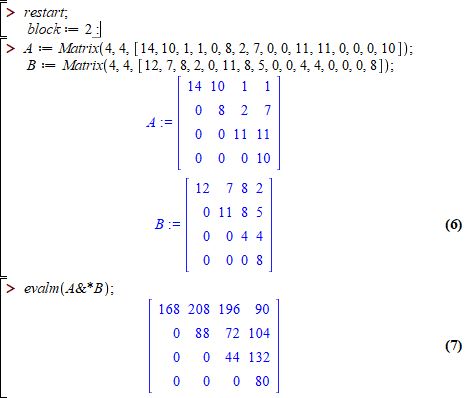
**Алгоритм решения**

Блочное умножение матриц – умножение строки блоков матрицы А на столбец блоков матрицы В.

Для блочного перемножения матриц понадобится шесть вложенных циклов for. Первые три цикла нужны для перемножения блоков матриц, а вторые три цикла для перемножения элементов внутри каждого блока. При переходе к новой строке матрицы A, положение указателя в матрице B возвращается в начало. На каждом шаге итерации происходит умножение блоков и прибавление полученного блока к соответствующему блоку в результирующей матрице Res. Указатель на блок в результирующей матрице Res сдвигается при проходе одного столбца в матрице B. Таким образом, используя вложенные циклы, получим блочное перемножение двух матриц.

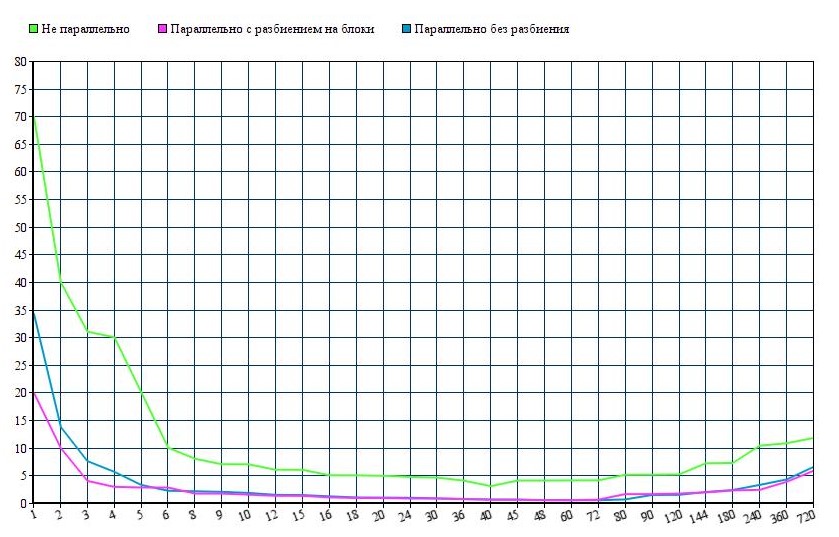
**Проверка работы программы**

Прежде чем перемножать матрицы больших размеров, проверим правильность работы алгоритма перемножения. Для этого перемножим матрицы размера 4х4 и убедимся, что программа работает правильно. Приведен правильный пример умножения матриц на Maple17.



**Полученные результаты**

Далее выясним, какой алгоритм работает эффективнее. Для этого сравним два разных алгоритма. Возьмём матрицы размера 720х720. Для наглядности построим график зависимости времени выполнения перемножения от размера блока.



**Характеристики компьютера**

Вычисления производились на компьютере со следующими характеристиками:

Процессор Intel Pentium 2020M  
Количество ядер процессора 2  
Частота 2.40 ГГц  
Кэш L2 512 Кб  
Размер оперативной памяти 4 ГБ

**Вывод**

Эксперимент показал, что распараллеливание даёт видимое улучшение по времени в случае, когда блоки в среднем имеют размер от 16 до 48. В других случаях тратится больше времени на распараллеливание, поэтому использование параллельных вычислений в матрицах с маленькими блоками или очень большими не оправдано. Самые неоптимальные варианты - разбиения матрицы на блоки размером 1x1 и 720x720. Такой результат обосновывается особенностью работы с кэш-памятью: при наилучшем разбиении, количество кэш-промахов уменьшается. Как следствие, получаем более быструю работу программы.