МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Нижегородский государственный университет им. Н.И. Лобачевского»**

**Национальный исследовательский университет**

**Институт информационных технологий, математики и механики**

**Кафедра математического обеспечения и суперкомпьютерных технологий**

**ЛАБОРАТОРНАЯ РАБОТА**

**«Умножение плотных матриц. Алгоритм Кэннона»**

**Выполнил: студент группы 1506-3**

**Тараканов Кирилл Сергеевич**

**\_\_\_\_\_\_\_\_\_\_\_ Подпись**

**Руководитель:**

**Сысоев Александр Владимирович**

**\_\_\_\_\_\_\_\_\_\_\_ Подпись**

**Нижний Новгород**

**2018**

Оглавление

[Постановка учебно-практической задачи 3](#_Toc514611781)

[Алгоритм умножения плотных матриц 4](#_Toc514611782)

[Алгоритм Кэннона 5](#_Toc514611783)

[Подтверждение корректности и генерация тестовых данных 7](#_Toc514611784)

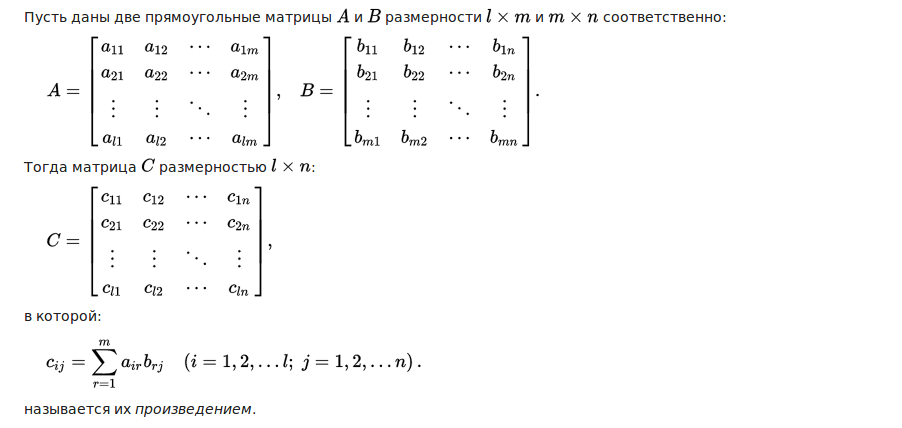
[Результаты экспериментов по оценке масштабируемости 8](#_Toc514611785)

# Постановка учебно-практической задачи

Целью данной лабораторной работы является написание программы, способной осуществлять умножение плотных матриц, заполненных элементами типа double, по алгоритму Кэннона. Программа будет включать в себя два варианта реализации, разработанных с помощью технологий параллельных вычислений для систем с общей памятью OpenMP и TBB (Threading Building Blocks). Кроме того, необходимым является наличие классического алгоритма умножения, который будет формировать эталонный результат, для подтверждения корректности работы OpenMP и TBB версий алгоритма Кэннона.

# Алгоритм умножения плотных матриц

Алгоритм перемножения матриц заключается в следующем:

**

Производим умножение матрицы размером N\*N на матрицу размером N\*N Пусть матрица хранится в двумерном массиве **int \*\*matrix**, и доступ к элементам осуществляется двойным индексом **matrix[i][j].**

Тогда реализация алгоритма умножения будет выглядеть следующим образом.

for (i = 0; i < n1; i++)

{

for (j = 0; j < m2; j++)

{

for (k = 0; k < m1; k++)

res[i][j] += (matrix1[i][k] \* matrix2[k][j]);

}

}

# Алгоритм Кэннона

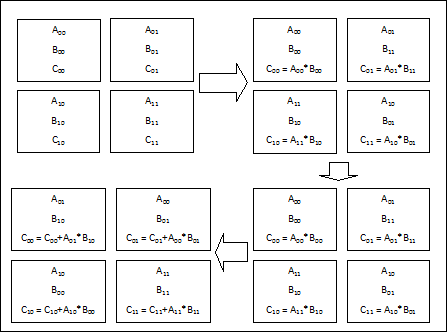
Алгоритма Кэннона — блочный. В блочном алгоритме матрицы разбиваются на блоки, представляющие собой подматрицы исходных матриц. При этом каждый блок Сij матрицы C определяется как произведение соответствующих блоков матриц A и B. Идея алгоритма состоит в изменении схемы начального распределения блоков перемножаемых матриц между процессорами вычислительной системы. Начальное расположение блоков в алгоритме Кэннона подбирается таким образом, чтобы располагаемые блоки на процессорах могли бы быть перемножены без каких-либо дополнительных передач данных между процессорами. При этом подобное распределение блоков может быть организовано таким образом, что перемещение блоков между процессорами в ходе вычислений может осуществляться с использованием более простых коммуникационных операций. Этап инициализации алгоритма Кэннона включает выполнение следующих операций передач данных:

1. На каждый процессор pij передаются блоки Aij, Bij;
2. Для каждой строки i процессорной решетки блоки матрицы A сдвигаются на (i-1) позиций влево;
3. Для каждого столбца j процессорной решетки блоки матрицы B сдвигаются на (j-1) позиций вверх.

В первую очередь создается некоторое число Р исполняющих процессов. Это число должно быть полным квадратом. Процессы организуются в виртуальную декартову топологию. Исходные матрицы должны иметь размерность, кратную квадратному корню из P. Основной этап алгоритма Кэннона представляет из себя цикл, в котором производятся вычисления блоков матрицы С, а основные коммуникационные операции: циклический сдвиг влево блоков матрицы A и циклический сдвиг вверх блоков матрицы В. В итоге алгоритм Кэннона выглядит следующим образом:

1. Для всех строк, кроме первой, производится циклический сдвиг влево блоков матрицы A.
2. Для всех столбцов, кроме первого, производится циклический сдвиг вверх блоков матрицы В.
3. Вычисляется C(i,j) = C(i,j) + A(i,j) \* B(i,j).
4. Осуществляется цикл для i от 0 до n–1: для всех строк производится циклический сдвиг влево блоков матрицы A; для всех столбцов производится циклический сдвиг вверх блоков матрицы В; вычисляется C(i,j) = C(i,j) + A(i,j) \* B(i,j).

Демонстрация



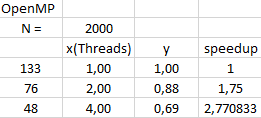
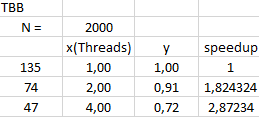
# Подтверждение корректности и генерация тестовых данных

Проверка корректности результатов осуществляется с помощью сравнения эталонного результата, полученного классическим алгоритмом умножения матриц, и результата одной из версий алгоритма Кэннона. Сравнение производится поэлементно. Для каждой пары элементов результирующих матриц вычисляется модуль разности, если он меньше заданной погрешности, то считается, что элементы пары равны между собой. Если модули для всех пар удовлетворяют этому неравенству, тогда считается, что матрицы равны между собой.

Для того, чтобы сгенерировать тестовые данные необходимо запустить генератор тестов, который сгенерирует две умножаемые матрицы заданного размера и заполнит их элементами типа double из диапазона -100.00 до 100.00.

# Результаты экспериментов по оценке масштабируемости

По результатам времени работы программы в зависимости от количества потоков был построен график scaling efficiency для OpenMP и TBB версий.

*** ***

Из представленных данных можно видеть, что за счет увеличения количества потоков при заданном размере матриц (N=2000) происходит уменьшение времени работы программы, но прирост ускорения является, при переходе от 2 к 4 потоком, небольшим, что говорит об относительно хорошей масштабируемости.