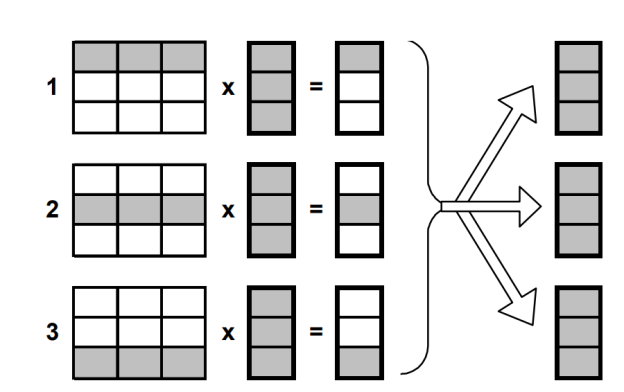
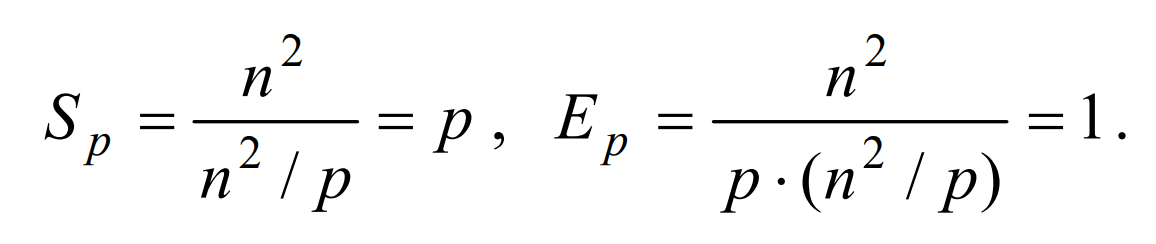
Работу выполнил Иослевич Григорий, 20 ПИ-1. Сделаны задания 1, 2

Задание 1

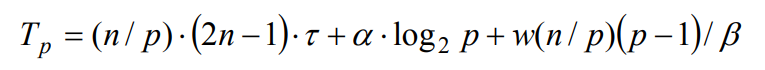
**Алгоритм 1 (строковый)**



S - ускорение, E - эффективность, p - количество процессов, n - количество строк

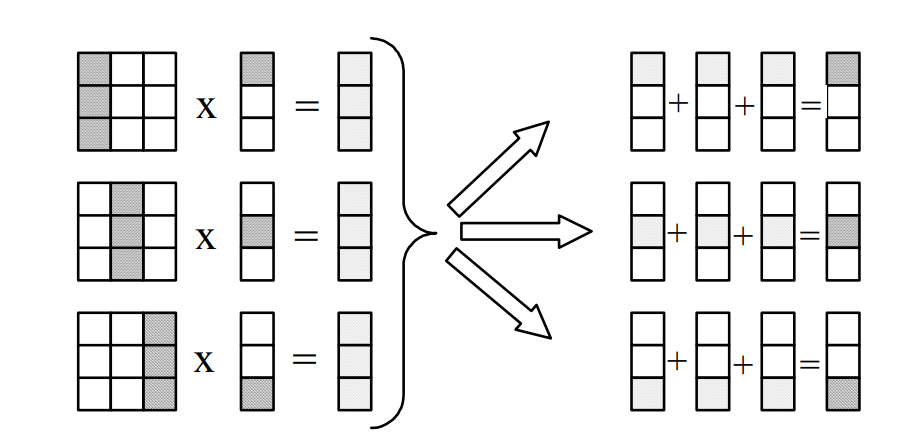


Время выполнения алгоритма:

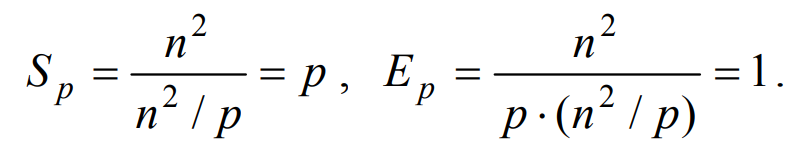


На вход алгоритм получает размерность матрицы, элементы матрицы, длину вектора, элементы вектора. Затем, алгоритм построчно умножает элементы матрицы на вектор. В зависимости от размерности матрицы и количества элементов алгоритм может работать сщественно быстрее или медленнее однопоточного аналога. Например, при умножении вектора на матрицу 8\*8m будет происходить false sharing, что снизит скорость работы многопоточной программы.

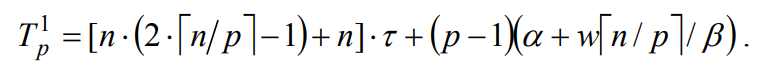
**Алгоритм 2 ( столбцовый)**



S - ускорение, E - эффективность, p - количество процессов, n - количество столбцов

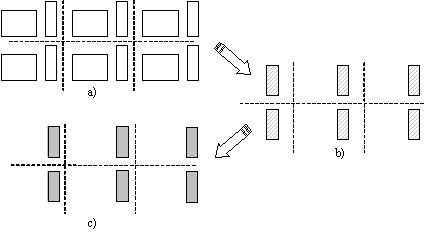


Время выполнения алгоритма:

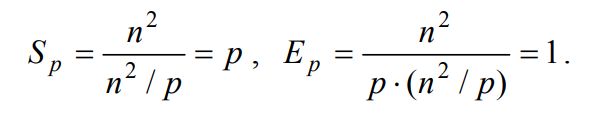


На вход алгоритм получает размерность матрицы, элементы матрицы, длину вектора, элементы вектора. Затем, алгоритм по столбцам умножает элементы матрицы на вектор. В зависимости от размерности матрицы и количества элементов алгоритм может работать сщественно быстрее или медленнее однопоточного аналога. Например, при умножении вектора на матрицу 8m\*8 будет происходить false sharing, что снизит скорость работы многопоточной программы.

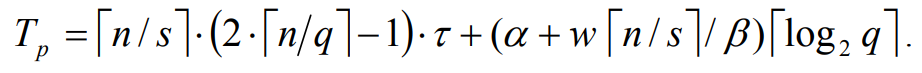
**Алгоритм 3 (Блочный)**



S - ускорение, E - эффективность, p - количество процессов

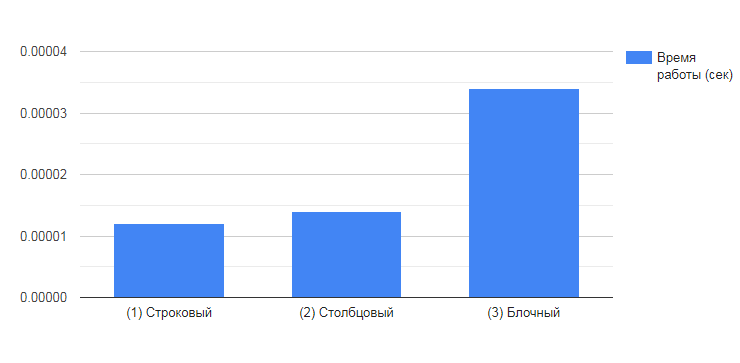


Время выполнения алгоритма:

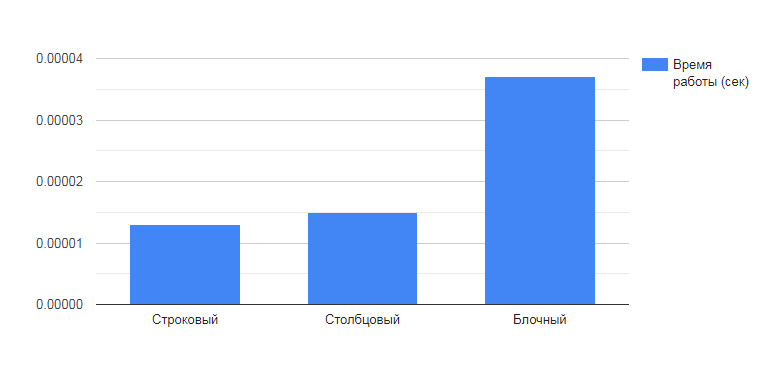


Графики работы алгоритмов (полные результаты работы лежат в файле results.txt)

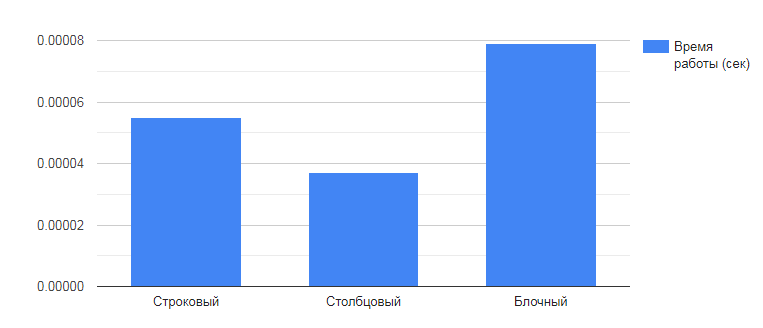
Матрица 3x3, вектор размера 3



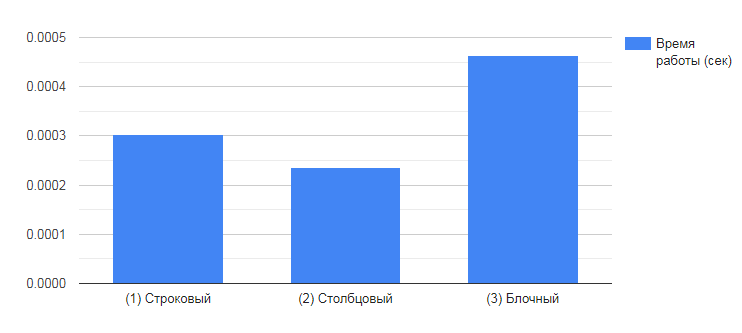
Матрица 30x30, вектор размера 30



Матрица 100x100, вектор размера 100



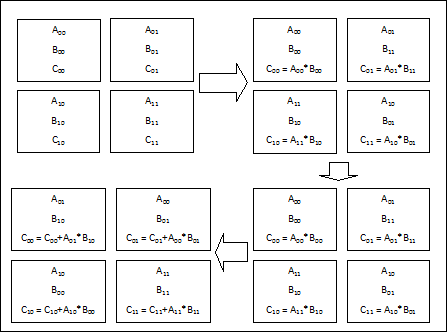
Матрица 300x300, вектор размера 300



Большее количество экспериментальных замеров находится в файле results.txt

Задание 2

Алгоритм Кэннона



Ускорение по сравнению с однопоточной программой:

nxn

Эффективность:

nxn

nxn