****МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

**Практикум по курсу**

**"Суперкомпьютеры и параллельная обработка данных"**

**Разработка параллельной версии программы для транспонирования матриц**

**ОТЧЕТ**

**о выполненном задании**

студентки 328 учебной группы факультета ВМК МГУ

Шелепнёвой Дарьи Дмитриевны

Москва, 2021 г.

Оглавление

[1 Постановка задачи - 2 -](#_Toc87626886)

[2 Описание алгоритма транспонирования матрицы - 2 -](#_Toc87626887)

[2.1 Основа: последовательный алгоритм - 2 -](#_Toc87626888)

[2.2 Параллельный алгоритм - 3 -](#_Toc87626889)

[3 Результаты замеров времени выполнения - 3 -](#_Toc87626890)

[3.1 Таблицы - 4 -](#_Toc87626891)

[3.2 Графики - 4 -](#_Toc87626892)

[4 Анализ результатов - 5 -](#_Toc87626893)

[5 Выводы - 5 -](#_Toc87626894)

# Постановка задачи

Ставится задача транспонирования матрицы.

Дана матрица , требуется получить матрицу , где .

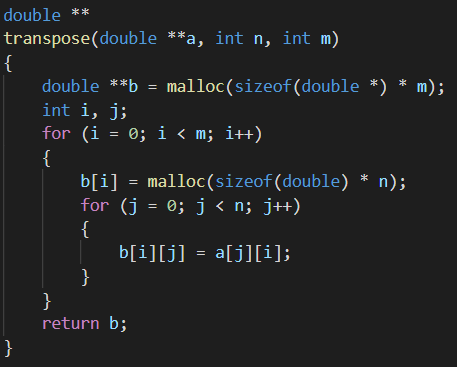
Требуется:

1. Реализовать параллельный алгоритм транспонирования матрицы с помощью технологии параллельного программирования OpenMP.
2. Исследовать масштабируемость полученных программ и построить графики зависимости времени выполнения программ от числа используемых потоков и объёма входных данных.

# Описание алгоритма транспонирования матрицы

## Основа: последовательный алгоритм

Простейший алгоритм транспонирования матрицы имеет следующий вид:

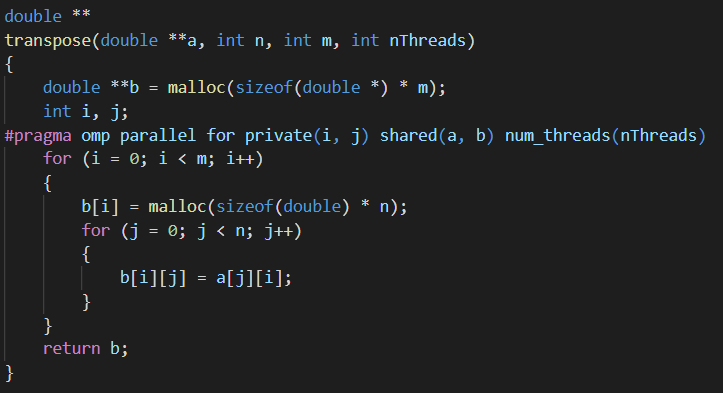


Этот алгоритм имеет сложность .

## Параллельный алгоритм

Разбиваем задачу вычисления конечной матрицы на подзадачи по вычислению строк и распределяем их по потокам. Разбиение на вычисление отдельных полей не производим, т.к. размеры матриц при вычислениях и так будут на порядки превышать число потоков.

В OpenMP модификация кода сводится к добавлению ***omp parallel for.***



Код всей программы можно посмотреть по ссылке <https://github.com/DariaShel/skipod/blob/main/transpose.c>

# Результаты замеров времени выполнения

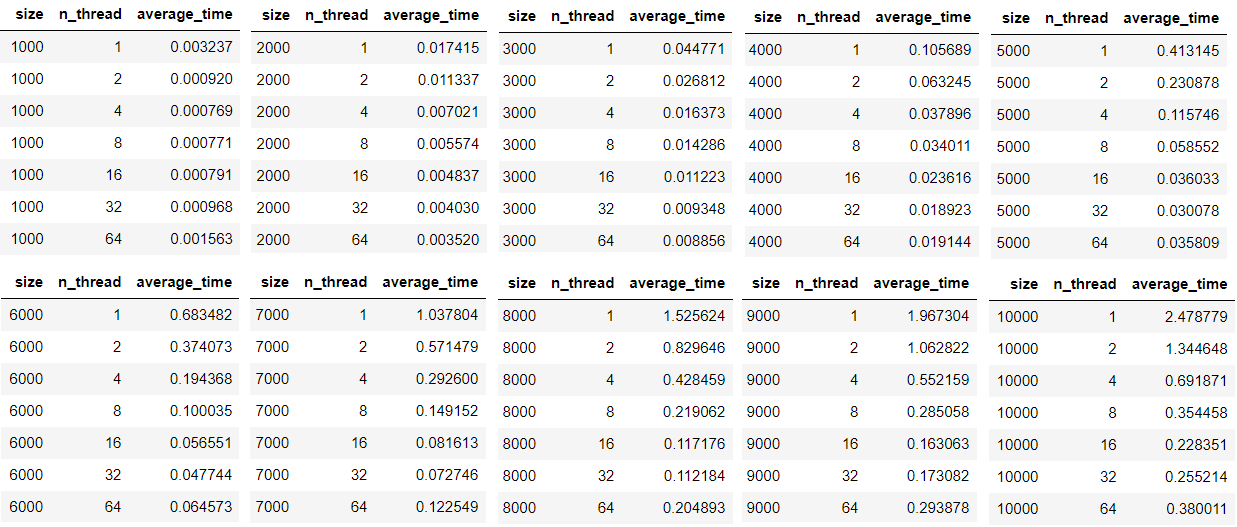
Ниже приведены результаты замеров времени выполнения алгоритма на суперкомпьютере Polus в табличной форме и наглядно на графиках.

Программа запускалась со следующими параметрами:

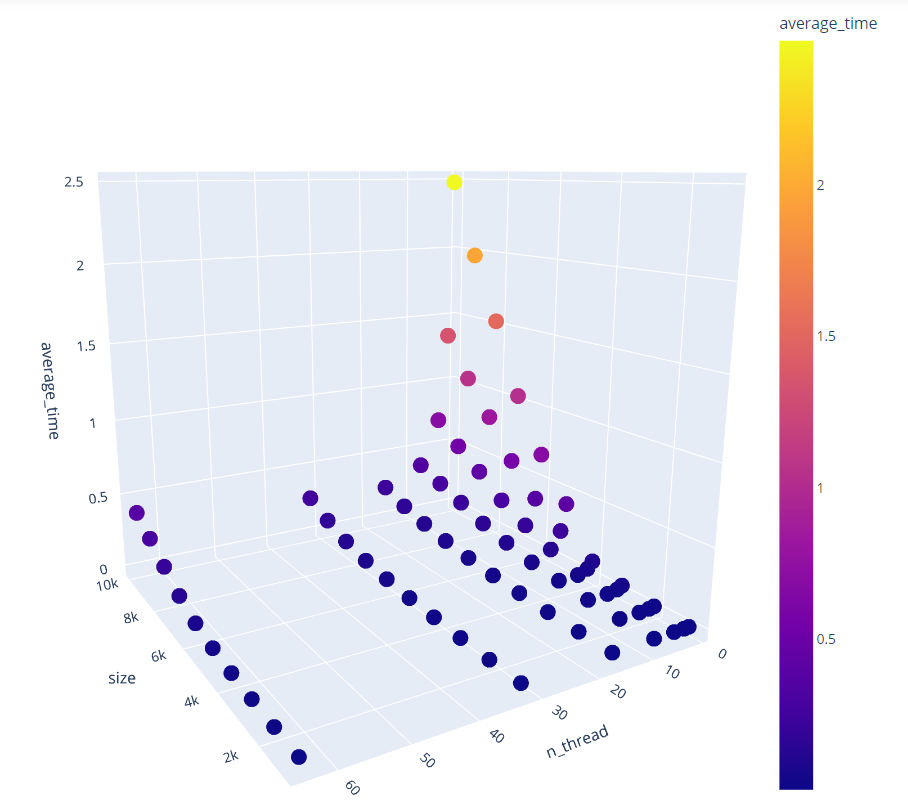
* Матрица
* Количество потоков

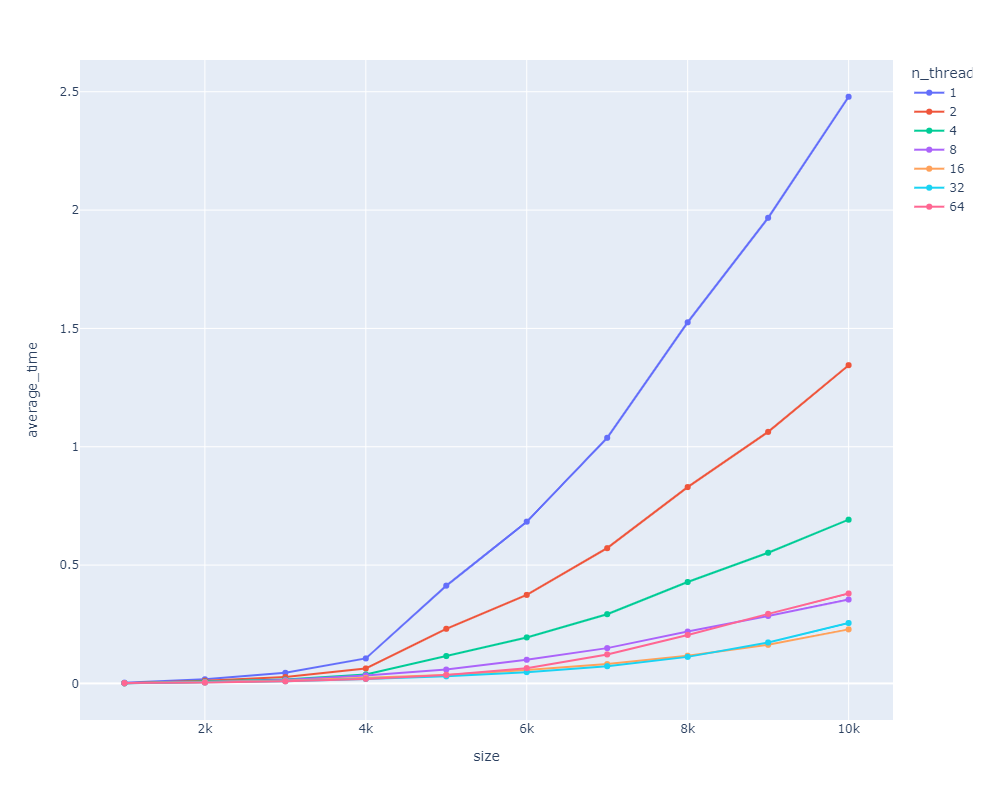
Было взято среднее значение времени за 5 запусков для каждой конфигурации.

## Таблицы



## Графики





# Анализ результатов

Из графиков и таблицы можно сделать вывод, что при увеличении количества потоков, особенно на больших размерах матриц, скорость выполнения алгоритма заметно увеличивается. Однако лучший результат получается распараллеливанием не на максимальное число потоков, а на 16. Это происходит из-за того, что при слишком большом количестве потоков тратятся ресурсы на управление ими, что приводит к дополнительным затратам времени.

# Выводы

Выполнена работа по разработке параллельной версии алгоритма транспонирования матриц. Изучена технология написания параллельного алгоритма OpenMP. Проанализировано время выполнения алгоритмов на вычислительной системе Polus.

Технология OpenMP достаточно удобна в использовании и даёт значительный прирост производительности на рассчитанных на многопоточные вычисления системах.