

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет информационных технологий  
Кафедра параллельных вычислений

Основы параллельного программирования

## Отчет

О выполнении работы № 3

**Работу**  
**выполнил:**  
Е. И. Биточкин  
Группа: 22209  
**Преподаватель:**  
А. А. Артюхов

Новосибирск  
2024

# Содержание

<b>1. Цель</b>	<b>3</b>
<b>2. Задание</b>	<b>3</b>
<b>3. Описание работы</b>	<b>4</b>
3.1. Реализация . . . . .	4
3.2. Профилирование . . . . .	5
3.3. Замеры . . . . .	7
3.3.1. Зависимость времени выполнения от числа процессов . . . . .	7
3.3.2. Зависимость времени выполнения от топологии сети . . . . .	9
3.3.3. Зависимость времени выполнения от размера матрицы . . . . .	10
<b>4. Заключение</b>	<b>11</b>
<b>5. Приложение</b>	<b>11</b>
5.1. Исходный код . . . . .	11

## 1. Цель

- Реализовать умножение матрицы на матрицу используя MPI
- Исследовать разработанный алгоритм

## 2. Задание

- Реализация алгоритма
- Выявления зависимости времени работы алгоритма от размера входных данных
- Выявления зависимости времени работы алгоритма от топологии сети
- Выявления зависимости времени работы алгоритма от числа процессов

## **3. Описание работы**

### **3.1. Реализация**

Алгоритм был реализован на C++. В качестве системы сборки был выбран CMake.

## 3.2. Профилирование

На рисунке 3.1 можно увидеть процесс рассылки частей матрицы между процессами.

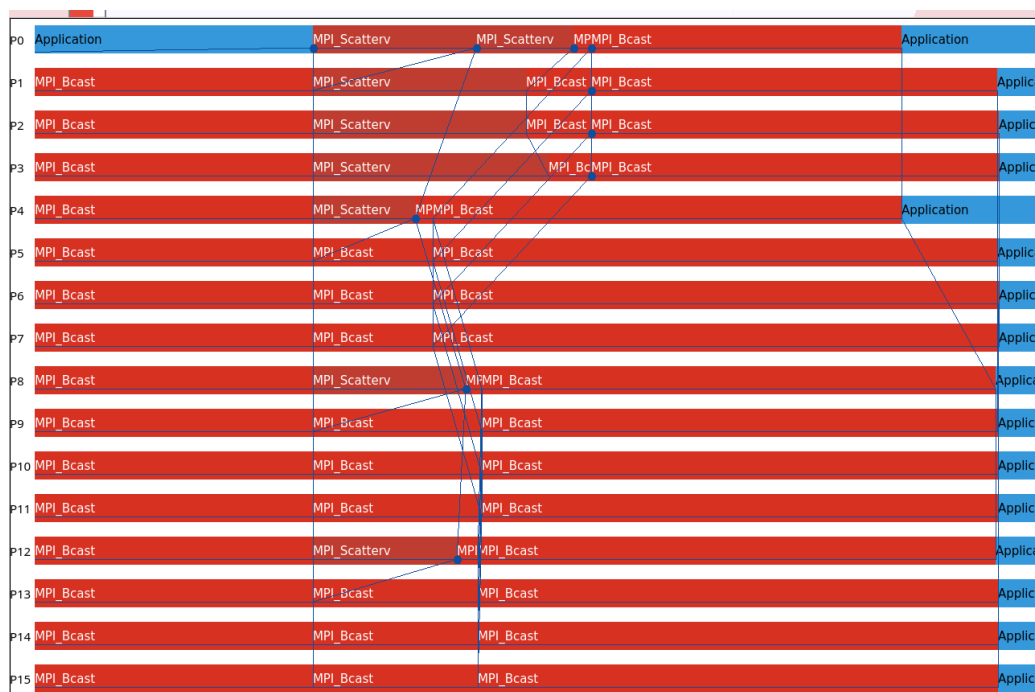


Рисунок 3.1. Рассылка матрицы между процессами

Линия в самом начале - отправка размеров матрицы между процессами через широковещательную рассылку. Далее основной процесс после чтения матрицы рассылает ее сначала по линиям, а затем по колонкам.

Однако, сама рассылка матрицы занимает много времени относительно общего времени (Рис 3.2):

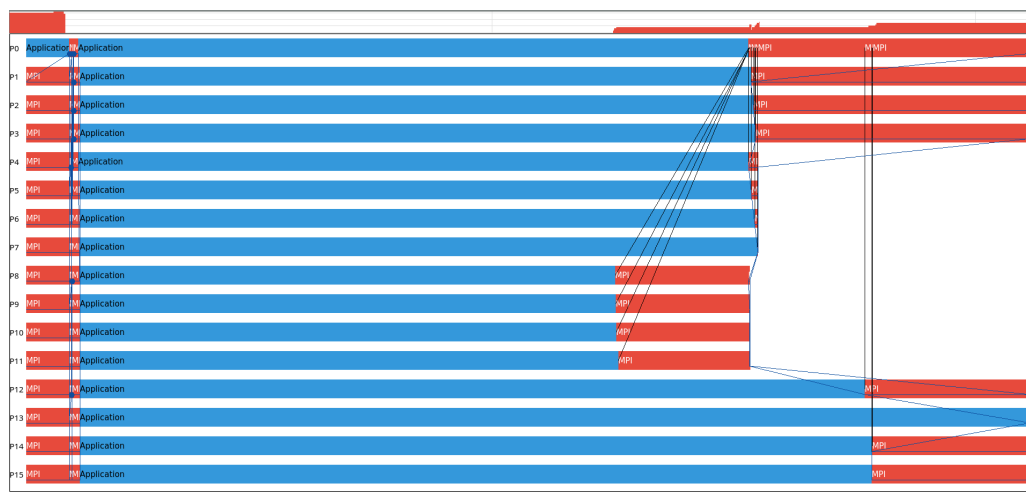


Рисунок 3.2. Полная картина

В конце можно увидеть сбор матрицы и замереного времени в корневом процессе (Рис 3.3):

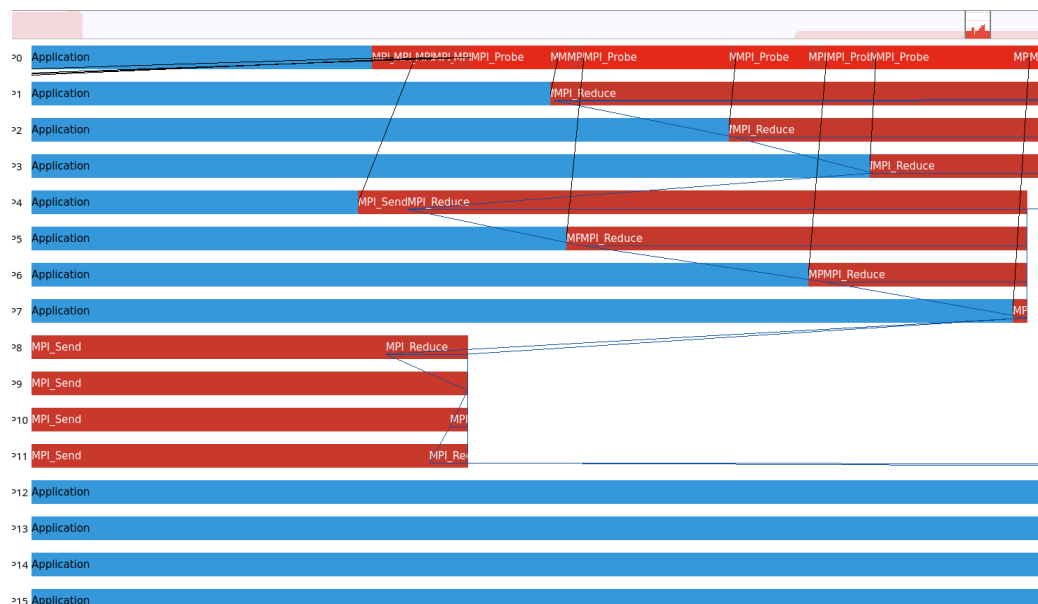


Рисунок 3.3. Сбор матрицы

На рисунке видна отправка из 8-11 процессов. Нулевой процесс через MPI\_Prob вычисляет позицию подматрицы и вставляет ее в результирующую матрицу. MPI\_Reduce - считает время.

### 3.3. Замеры

#### 3.3.1. Зависимость времени выполнения от числа процессов

На кластере был произведен замер для выявления зависимости времени работы от числа процессов:

Процессов:	Время	Процессов:	Ускорение:	Процессов:	Эффективность:
1	1266,3	1	1	1	100
2	791,338	2	1,6	2	80
4	491,625	4	2,576	4	64,4
8	235,108	8	5,386	8	67,33
16	213,01	16	5,945	16	37,16

Рисунок 3.4. Время в секундах, Ускорение в долях единиц, Эффективность в процентах

Все вышеописанные замеры были сделаны на одинаковых (матрица  $10K \times 10K$  умножалась на матрицу тех же размеров). Топология в каждом запуске была выбрана по умолчанию.

Была выявлена зависимость времени исполнения от числа процессов:

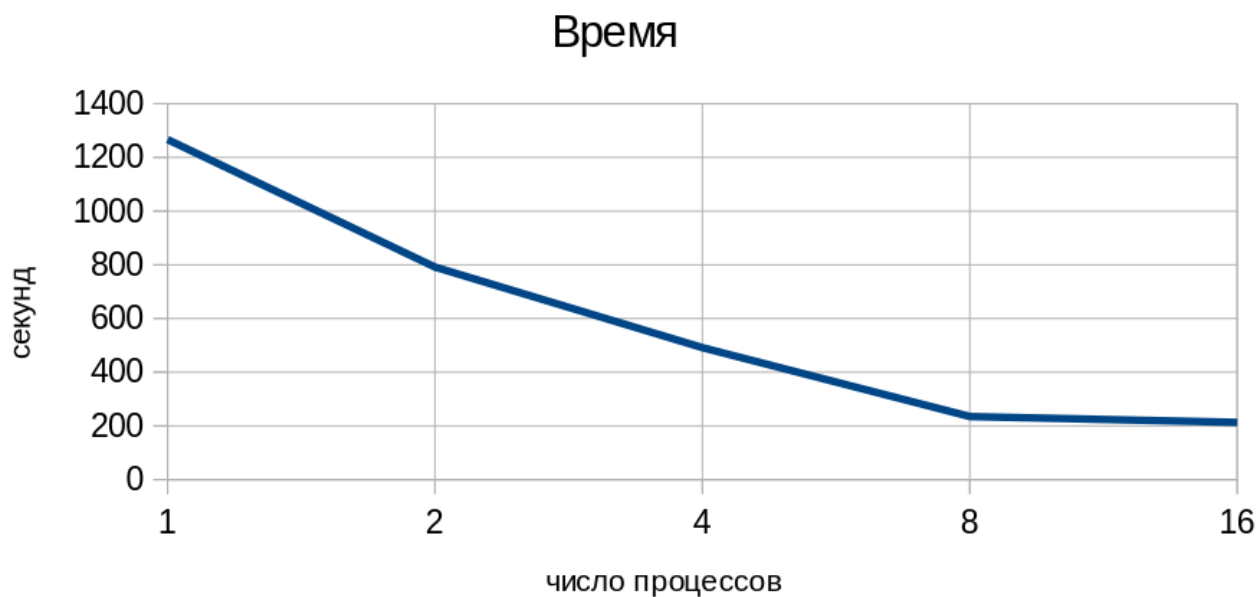
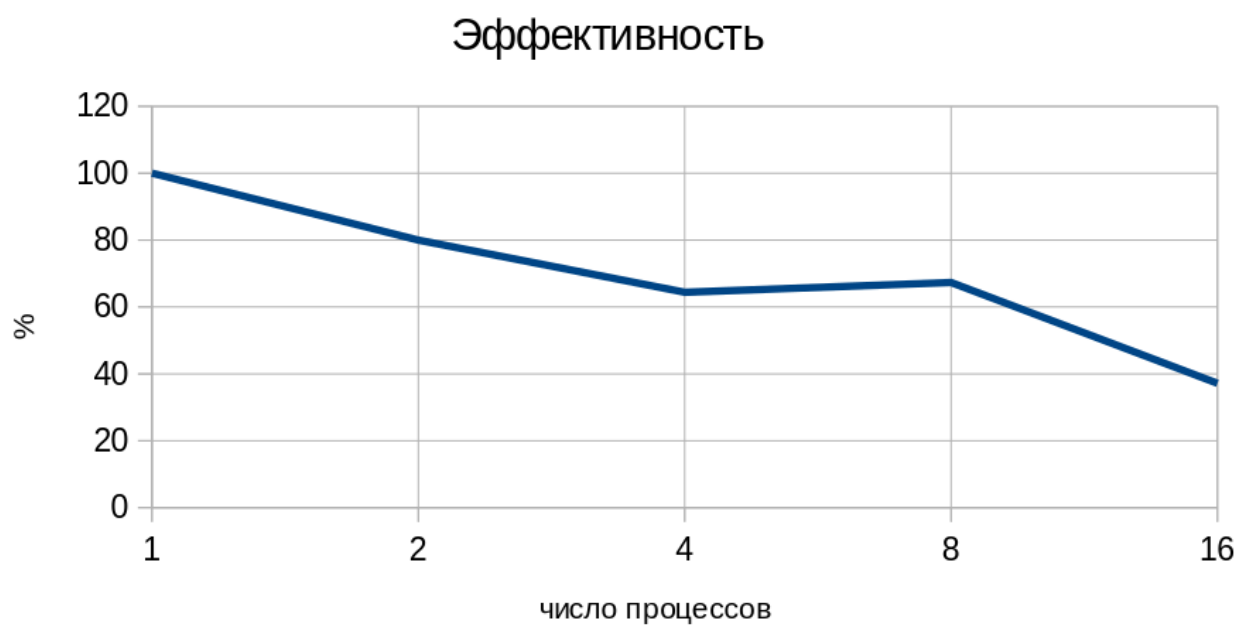
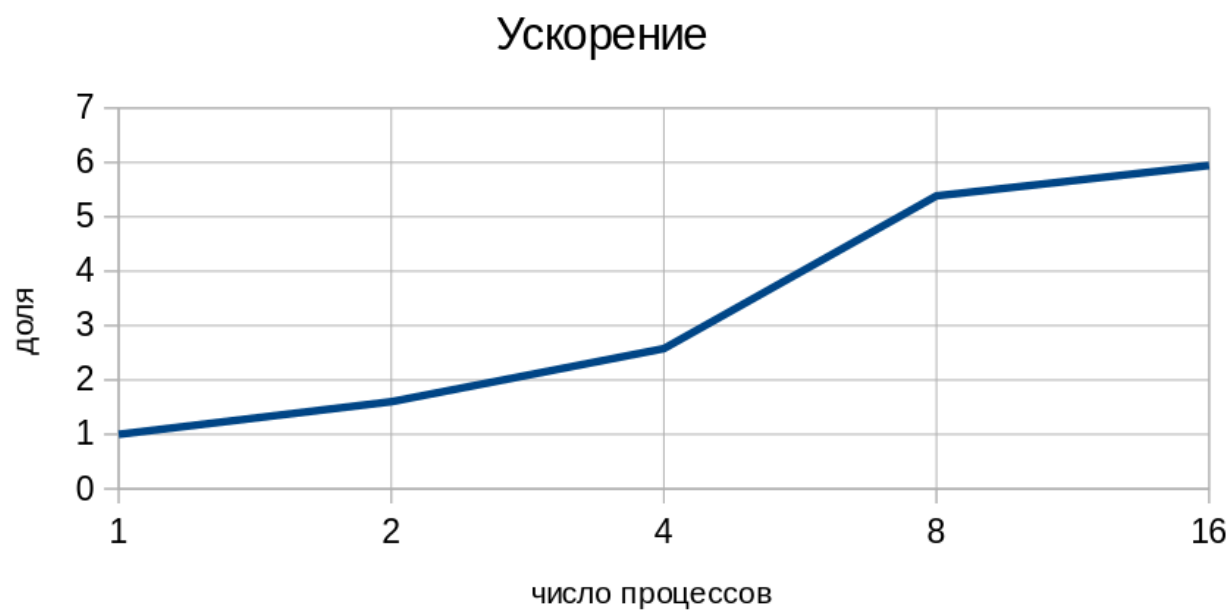


Рисунок 3.5. Enter Caption





### 3.3.2. Зависимость времени выполнения от топологии сети

Топология сети тестировалась на тех же данных, что и в пункте 3.3.1. Запуске проводились на шестнадцати процессах (и четырех узлах кластера, соответственно).

	А	В	С
1	Топология		Время
2	строк	столбцов	
3	1	16	262,012
4	2	8	290,587
5	4	4	288,879
6	8	2	286,973
7	16	1	285,983
8			

Рисунок 3.6. Красным - максимальное, фиолетовым - минимальное

Из полученных данных невозможно четко установить корреляцию между топологией и времени.

Среднее (секунд):
282,8868
Отклонение от среднего(%):
7,38
Максимальная разница (%):
10,1

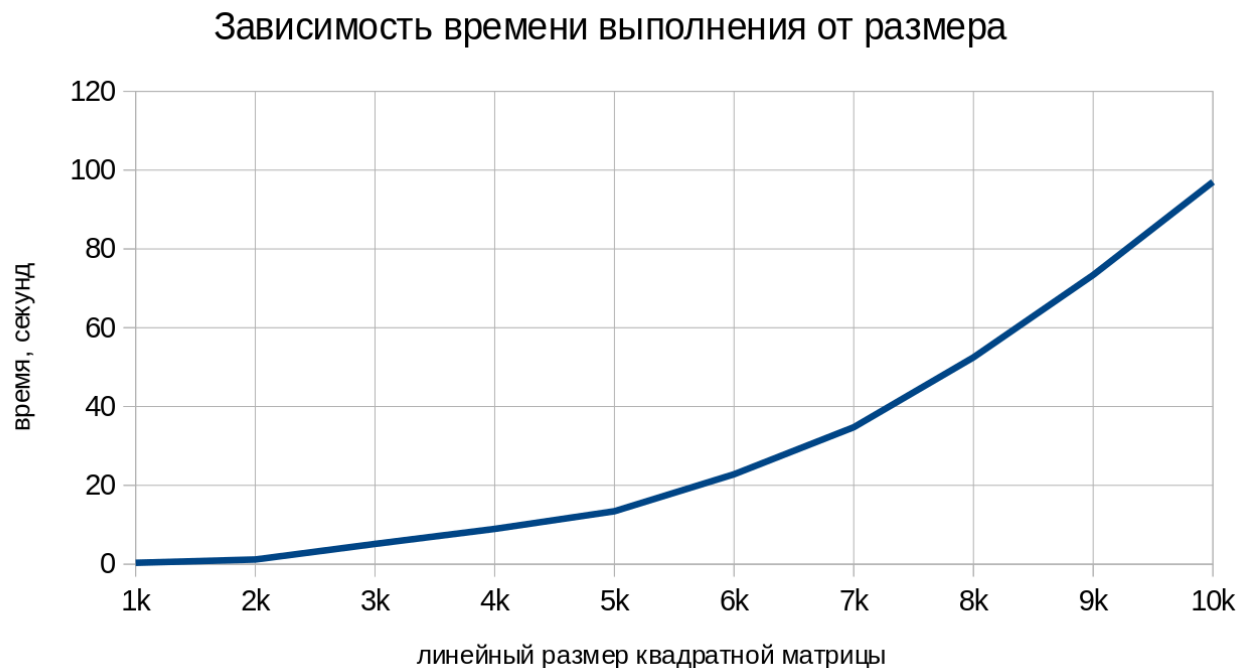
Рисунок 3.7. Отклонение

### 3.3.3. Зависимость времени выполнения от размера матрицы

Для исследования зависимости времени от размера матрицы была взята топология по умолчанию с шестнадцатью процессами. Входные данные - две единичные матрицы одинакового размера (размер указан в таблице).

	A	B
1	Размер матрицы	Время, секунд
2	1k	0,336393
3	2k	1,18609
4	3k	5,13834
5	4k	8,9443
6	5k	13,4363
7	6k	22,8195
8	7k	34,7548
9	8k	52,4747
10	9k	73,3727
11	10k	96,9922
12		

Рисунок 3.8. Результат замеров



Асимптотика умножения матриц:  $O(n^3)$ . Как видно, график приближенно напоминает график функции  $x^3$ .

## 4. Заключение

Был реализован алгоритм умножения матриц на нескольких процессах. Рассмотрена эффективность изложенного подхода.

- Зависимость времени работы алгоритма от числа процессов рассмотрена в пункте 3.3.1.
- Зависимость времени работы алгоритма от топологии сети не выявлена.
- Зависимость времени работы параллельного алгоритма такая же, как и у последовательного -  $O(n^3)$

## 5. Приложение

### 5.1. Исходный код

[https://github.com/BigCubeCat/bpp\\_labs/tree/master/matmult](https://github.com/BigCubeCat/bpp_labs/tree/master/matmult)