

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ
«МИФИ»
КАФЕДРА №42 «КРИПТОЛОГИЯ И КИБЕРБЕЗОПАСНОСТЬ»

ОТЧЁТ

по дисциплине «Параллельное программирование»
Лабораторная работа №5
«Технология MPI. Введение»

Группа

Б21-525

Студент

Г.О. Шулындин

Преподаватель

М.А. Куприяшин

Москва 2023

Оглавление

1.	Описание рабочей среды	3
2.	Open MPI	3
3.	OpenMP	4
4.	Заключение	6
5.	Приложение	6

1. Описание рабочей среды

- Модель процессора: AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx
- Число ядер: 8
- Архитектура: x86-64
- ОС: Linux, дистрибутив Ubuntu v22.04
- RAM объем: 2x4096 MB (В оперативной памяти есть проблемы. Требуется замена)
- RAM тип: DDR4
- Используемая среда разработки: Visual Studio Code
- Компилятор: gcc v11.4.0
- Поддерживаемая версия OpenMP: 201511
- Версия Open MPI: 4.0

2. Open MPI

Был реализован параллельный алгоритм поиска максимального элемента в массиве с использованием библиотеки Open MPI.

Описание эксперимента

- Измеряется время работы алгоритма на разном числе потоков: от 1 до 10.
- Измерения проводятся для 50 случайно сгенерированных массивов длиной 10 000 000 элементов.
- Находится среднее время работы для каждого числа потоков.

Построение линейного алгоритма

Был реализован линейный алгоритм поиска максимального элемента в массиве. Среднее время работы составляет 0.034598с для массива длиной 10 000 000 элементов.

Построение параллельного алгоритма

На основе линейного алгоритма был построен параллельный алгоритм поиска максимального элемента. Среднее время работы на каждом числе потоков:

Number of threads	Exec. time
1	0.034360
2	0.018255
3	0.013769
4	0.010549
5	0.007627
6	0.006645
7	0.006934
8	0.015772
9	0.032957
10	0.046348

3. OpenMP

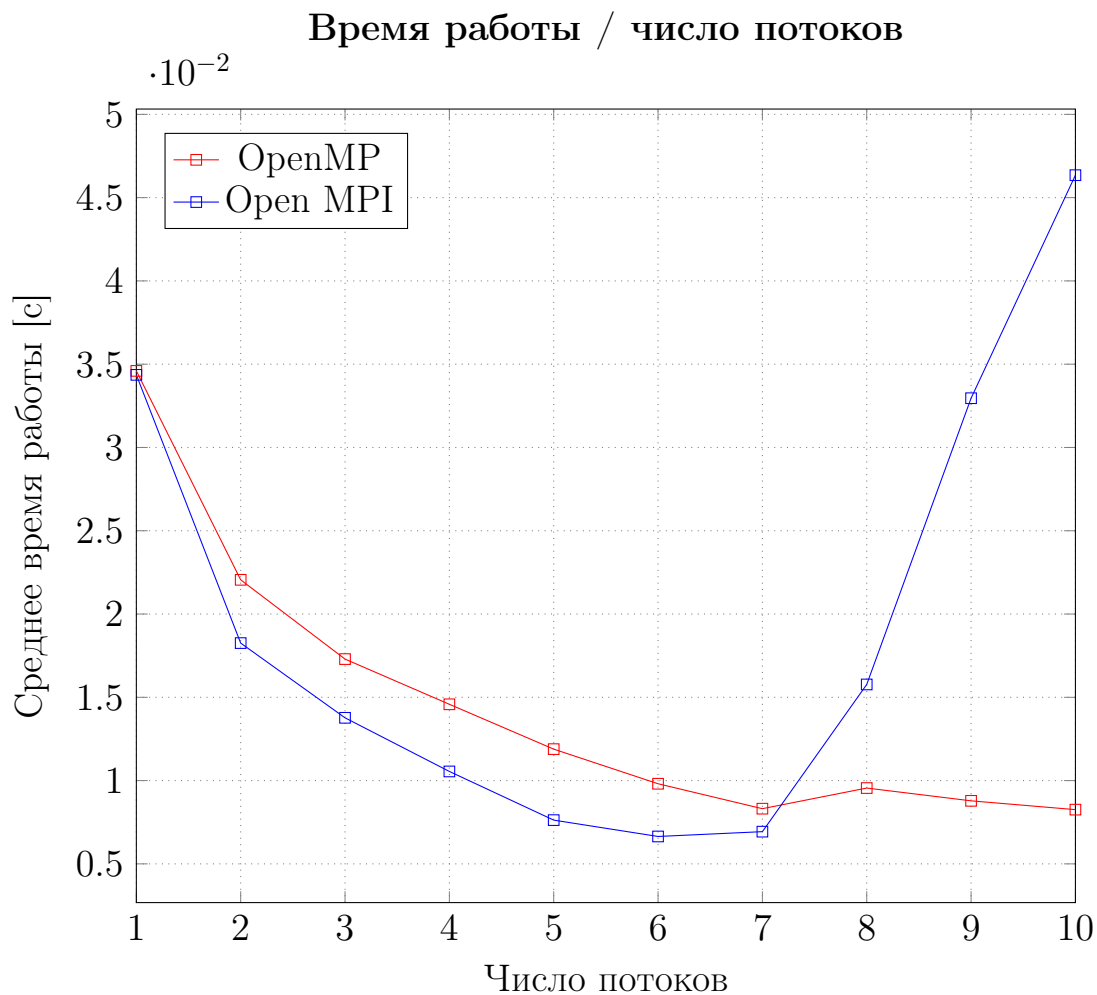
Для сравнения был взят параллельный алгоритм поиска максимального элемента в массиве из лабораторной работы №1.

Среднее время работы на каждом числе потоков:

Number of threads	Exec. time
1	0.034598
2	0.022054
3	0.017295
4	0.014581
5	0.011889
6	0.009810
7	0.008313
8	0.009552
9	0.008785
10	0.008256

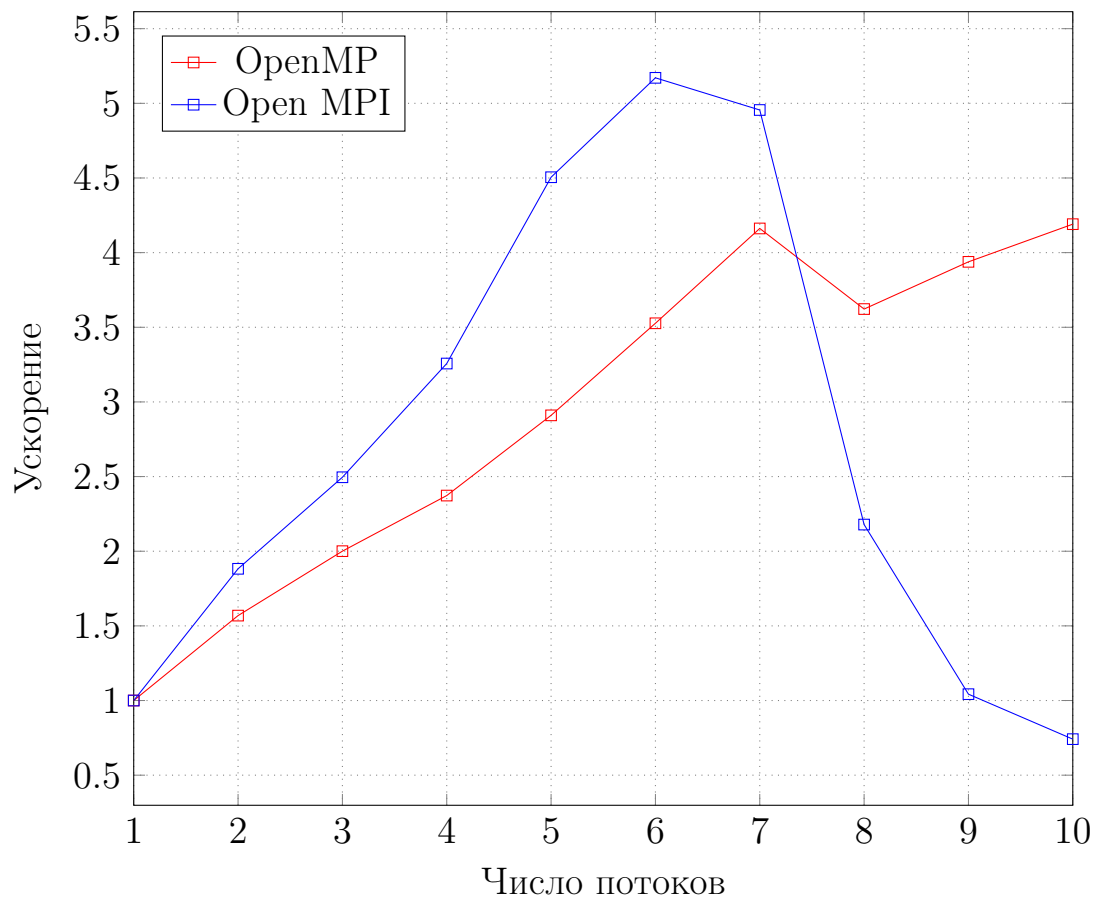
Графики

Среднее время работы



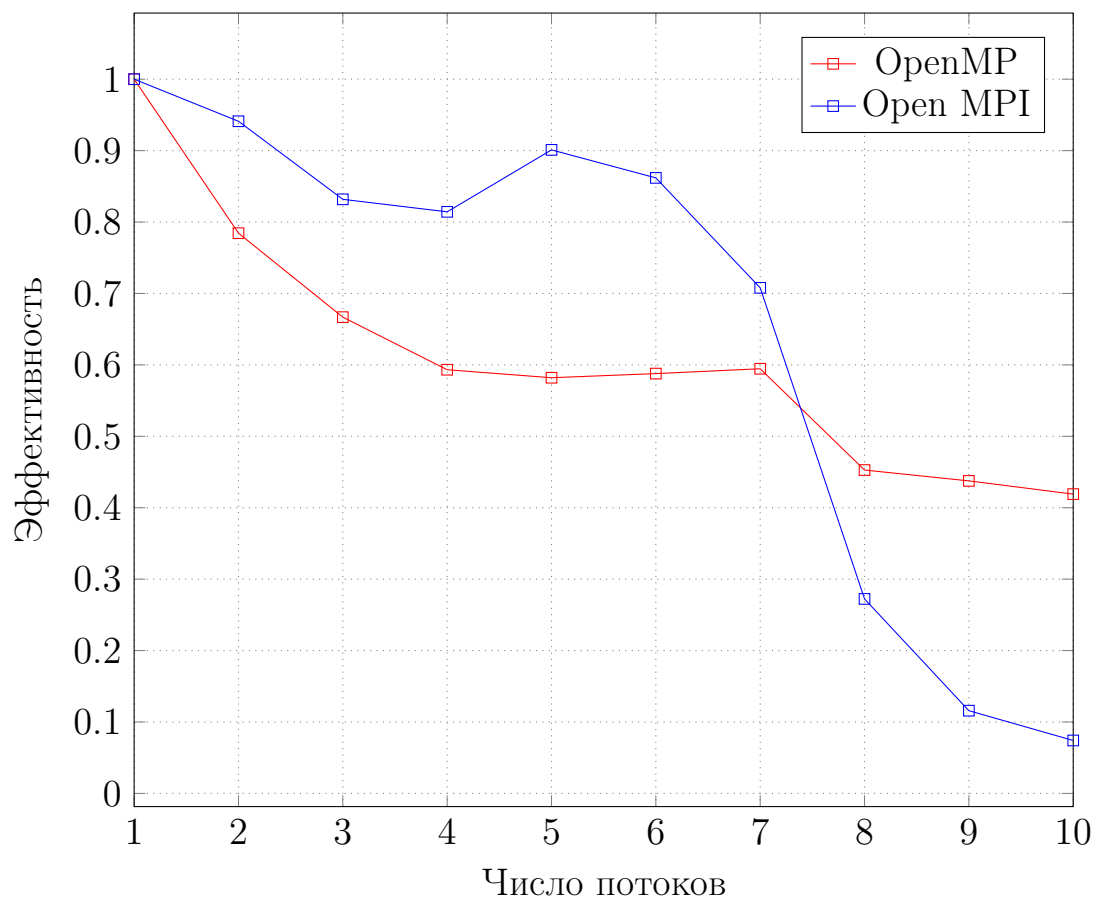
Ускорение

Ускорение / число потоков



Эффективность

Эффективность / число потоков



4. Заключение

В ходе лабораторной работы был реализован параллельный алгоритм поиска максимального элемента в массиве с использованием библиотеки Open MPI. Также был реализован тот же алгоритм, но с использованием технологии OpenMP. Было измерено среднее время работы алгоритмов на разном числе потоков. Были вычислены значения ускорения и эффективности, построены соответствующие графики.

На основе этих данных можно сделать следующие выводы:

- Для числа потоков от 1 до 7 время работы алгоритма с использованием Open MPI примерно равно времени работы алгоритма с использованием OpenMP. Максимальное в этом промежутке числа потоков ускорение при использовании Open MPI составляет 5.17 на 6 потоках; при использовании OpenMP 4.16 на 7 потоках.
- Для числа потоков больше 7 время работы при использовании OpenMP остается примерно одинаковым, а при использовании Open MPI начинает расти.

5. Приложение

Код программы расположен на github

Запуск программы: `make all`