МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Вятский государственный университет» Факультет автоматики и вычислительной техники Кафедра электронных вычислительных машин

**Алгоритм для перемножения полиномов с помощью быстрого преобразования Фурье**

Отчет по лабораторной работе №8

«Высокопроизводительные вычислительные комплексы»

Выполнил студент группы ИВТ-43\_\_\_\_\_\_\_\_\_\_\_\_\_/Кудяшев Я.Ю./

Проверил преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Мельцов В. Ю./

Киров 2022

1. Задание

Необходимо написать программу с использованием MPI для запуска на кластере (многоядерной ЭВМ), протестировать на различном количестве ядер и при различных размерностях данных, построить графики зависимостей времени от количества используемых ядер и от размерности данных.

1. Выполнение лабораторной работы

В качестве распараллеливаемой вычислительной задачи в данной лабораторной работе используется алгоритм для перемножения полиномов с помощью быстрого преобразования Фурье.

Быстрое преобразованье Фурье – алгоритм ускоренного вычисления дискретного преобразования Фурье, позволяющий получить результат за время, меньшее чем *O(N2)* (требуемого для прямого, поформульного вычисления).

Метод БФП основывается на свойствах комплексных корней из единицы: на том, что степени одних корней дают другие корни. Благодаря данному свойству этот метод позволяет вычислить ДПФ и ОПФ за время *O(n log n)*.

Из исследований алгоритма для перемножения полиномов с помощью быстрого преобразованья Фурье удалось выяснить, что время в большей степени зависит от количества входных векторов, нежели от размерности. Было принято решение переложить работу по умножению каждой пары векторов на потоки. Помимо этого, для более эффективного использования потоков, было сделано одно нововведение в алгоритм распараллеливания: пока идёт цикл перемножения векторов, главный поток подготавливает новый массив данных для следующего теста. Данный подход позволяет уменьшить общее время выполнения программы.

Таким образом, после перемножения пары векторов, каждое последующее умножение будет происходить уже с полученным в результате предыдущего умножения вектором.

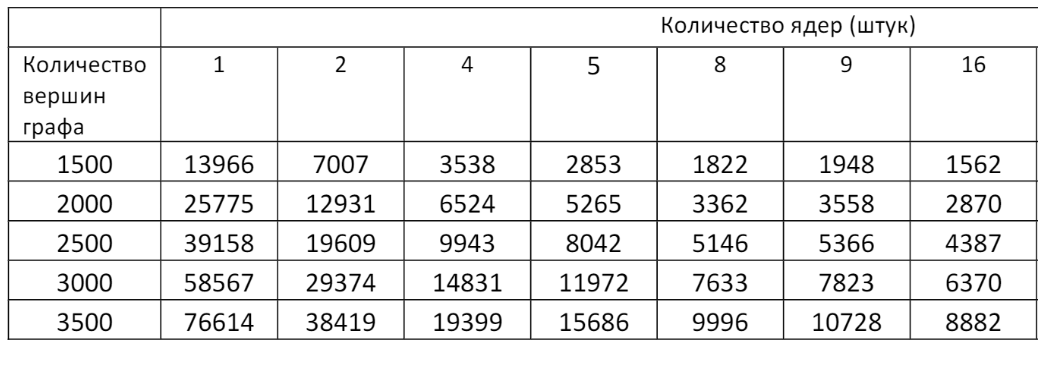
1. Тестирование

Тестирование проводилось на сети из 2-х ЭВМ. ЭВМ под управлением 64-разрядной OC Windows 10, с 8 ГБ оперативной памяти являлась главной в кластере. Для подключения 2-х ЭВМ в одну локальную сеть использовался сетевой мост, хост-системы подключались к общей Wi-Fi-сети. В общей сложности кластер состоял из 16 логических ядер.

В ходе тестирования использовались ЭВМ:

* Процессор Intel Core i5-8250U с частотой 1.80 ГГц (8 логических или 4 физических ядра);
* Процессор Intel Core i5-1135G7 (8 логических или 4 физических ядра).

В качестве исходных данных было принято решение взять 16 полиномов, размерность которых изменялось в диапазоне от 10 тыс. до 1 млн.

Таблица 1 – Результаты тестирования

Размерность векторов, тыс

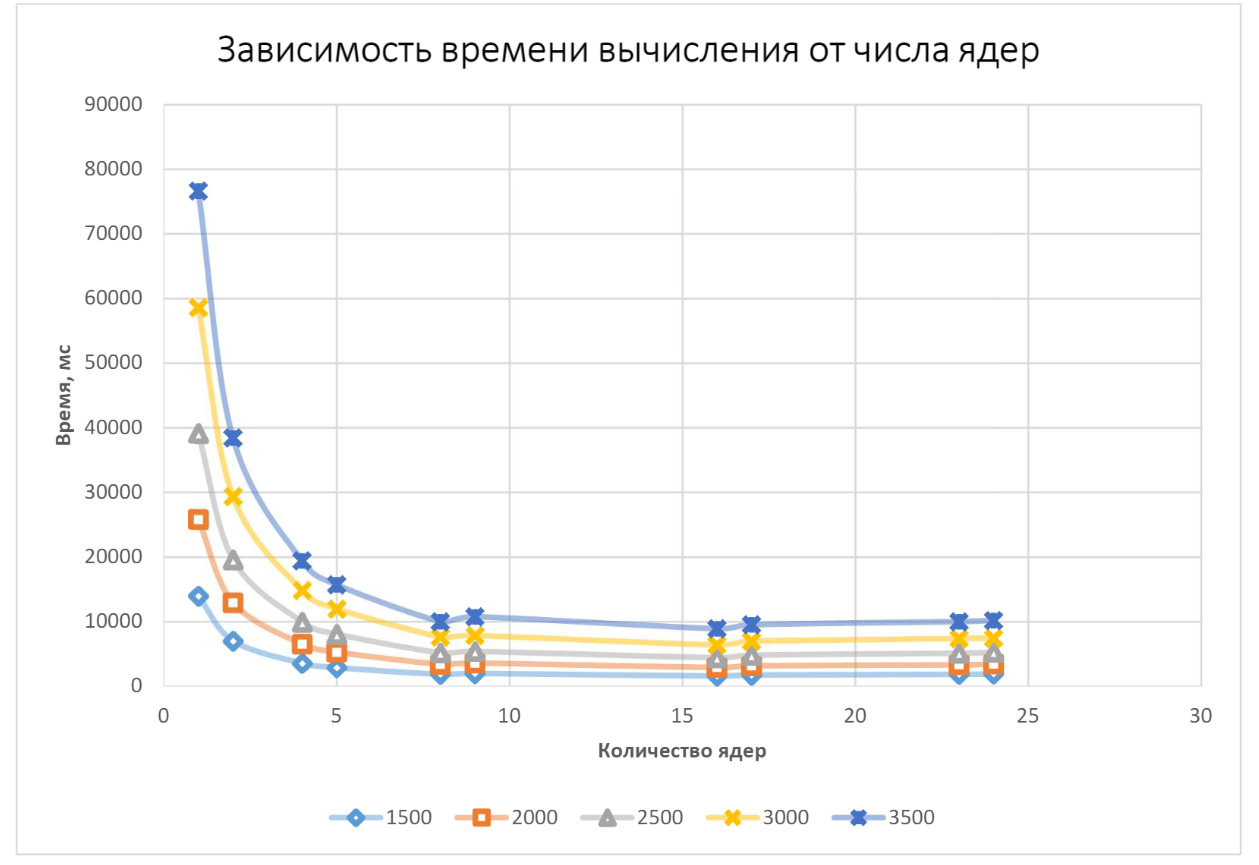


Рисунок 1 – Зависимость времени вычисления от числа ядер

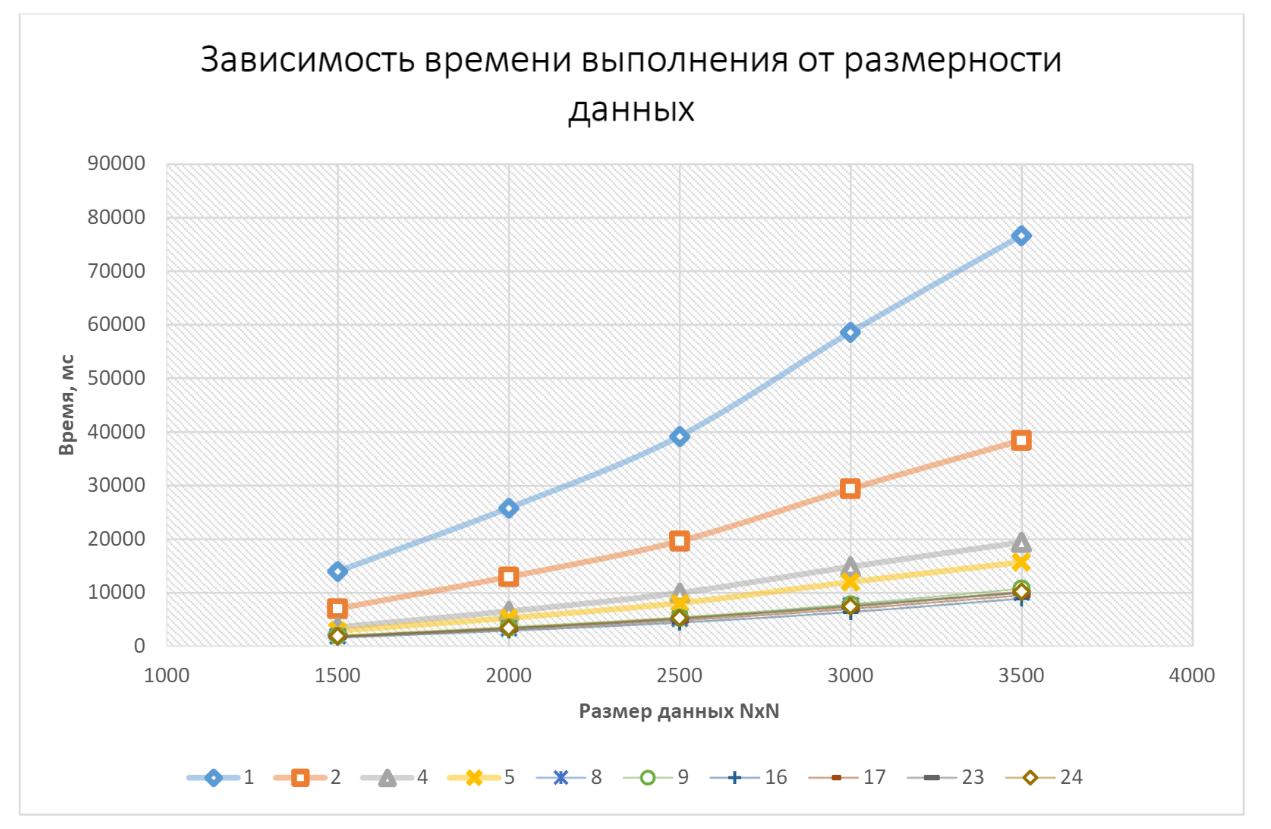


Рисунок 2 – Зависимость времени выполнения от размерности данных

1. Вывод

Анализируя данные в таблице 1 и графики можно сделать следующие выводы:

* До 8 ядер при неизменном количестве данных и увеличении числа используемых ядер время решения задачи наиболее быстро уменьшается. При количестве ядер больше 8 эффективность параллельных вычислений снижается из-за расхода времени на обмен данными между параллельными потоками.;
* При решении задачи на 9 и более ядрах происходит заметное уменьшение ускорения параллельного алгоритма, при этом, минимальное время решения задачи достигается на 16 ядрах. Разница в интенсивности изменения времени связана прежде всего с тем, что при малом количестве ядер время решения задачи в большей степени состоит из перемножения полиномов и в меньшей – на пересылку данных. В случае же большего количества ядер ситуация противоположная;
* При увеличении размерности данных и одинаковом количестве используемых ядер время решения увеличивается быстрее всего при использовании небольшого количества ядер;
* При одновременном равномерном увеличении количества используемых ядер и размерности данных время решения задачи увеличивается из-за дополнительных задержек на передачу данных и организацию вычислений.