Государственное образовательное учреждение высшего профессионального образования
“Московский государственный технический университет имени Н.Э.Баумана”

Дисциплина: Анализ алгоритмов

Лабораторная работа 4

Умножение матриц с помощью алгоритма Винограда с параллельными вычислениями.

Студент группы ИУ7-55,
Шестовских Николай Александрович

2019 г.

Содержание

Введение

Целью данной работы является изучение и реализация параллельного алгоритма Винограда для умножения матриц. Необходимо сравнить зависимость времени работы алгоритма от числа параллельных потоков исполнения и размера матриц, провести сравнение стандартного и параллельного алгоритма.

1 Аналитическая часть

В данной части будут рассмотрены теоретические основы алгоритмов.

1.1 Теоретические сведения об умножении матриц

Матрица – это прямоугольная таблица каких-либо элементов. Здесь и далее мы будем рассматривать только матрицы, элементами которых являются числа. Упорядоченная пара чисел (n, m), где n - количество строк в матрице, m - количество столбцов, называется размерностью матрицы, обозначается обычно m x n[1].

Пусть имеются две матрицы: A и B размерами n x l и l x m соответственно.

$$\begin{bmatrix} a_{1,1} & a_{1,2} & ... & a_{1,l} \\ a_{2,1} & a_{2,2} & ... & a_{2,l} \\ ... & ... & ... & ... \\ a_{n,1} & a_{n,2} & ... & a_{n,l} \end{bmatrix}$$

$$\begin{bmatrix} b_{1,1} & b_{1,2} & ... & b_{1,m} \\ b_{2,1} & b_{2,2} & ... & b_{2,m} \\ ... & ... & ... & ... \\ b_{l,1} & b_{l,2} & ... & b_{l,m} \end{bmatrix}$$

Произведением матриц A и B размерами n x l и l x m соответственно называется матрица C размерами n x m, каждый элемент которой вычисляется по формуле (1):

$$c_{i,j} = \sum_{r=1}^{n} a_{i,r} \cdot b_{r,j} \tag{1}$$

$$\begin{bmatrix} c_{1,1} & b_{1,2} & ... & c_{1,m} \\ c_{2,1} & b_{2,2} & ... & c_{2,m} \\ ... & ... & ... & ... \\ c_{n,1} & c_{n,2} & ... & c_{n,m} \end{bmatrix}$$

1.2 Алгоритм Винограда

Если посмотреть на результат умножения двух матриц, то видно, что каждый элемент в нем представляет собой скалярное произведение соответствующих строки и столбца исходных матриц. Также некоторые вычисления можно произвести заранее, что ускорит выполнение алгоритма. Рассмотрим два вектора V = (v_1, v_2, v_3, v_4) и W = (w_1, w_2, w_3, w_4)
Их скалярное произведение находится по формуле (2)

$$V \cdot W = v_1 \cdot w_1 + v_2 \cdot w_2 + v_3 \cdot w_3 + v_4 \cdot w_4 \tag{2}$$

Равенство (2) можно переписать в виде (3)

$$V \cdot W = (v_1 + w_2) \cdot (v_2 + w_1) + (v_3 + w_4) \cdot (v_4 + w_3) - v_1 \cdot v_2 - v_3 \cdot v_4 - w_1 \cdot w_2 - w_3 \cdot w_4 \tag{3}$$

В Алгоритме Винограда используется скалярное произведение из формулы 2, в отличие от стандартного алгоритма. Алгоритм Винограда позволяет выполнить предварительную обработку матрицы и запомнить значения для каждой строки/столбца матриц. Над предварительно обработанными элементами нам придется выполнять лишь первые два умножения и последующие пять сложений, а также дополнительно два сложения[2].

1.3 Параллельные вычисления

Параллельные вычисления — способ организации компьютерных вычислений, при котором программы разрабатываются как набор взаимодействующих вычислительных процессов, работающих параллельно (одновременно).

При использовании многопроцессорных вычислительных систем с общей памятью обычно предполагается, что имеющиеся в составе системы процессоры обладают равной производительностью, являются равноправными при доступе к общей памяти, и время доступа к памяти является одинаковым (при одновременном доступе нескольких процессоров к одному и тому же элементу памяти очередность и синхронизация доступа обеспечивается на аппаратном уровне). Многопроцессорные системы подобного типа обычно именуются симметричными мультипроцессорами (symmetric multiprocessors, SMP).
Перечисленному выше набору предположений удовлетворяют также активно развиваемые в последнее время многоядерные процессоры, в которых каждое ядро представляет практически независимо функциони рующее вычислительное устройство.

Обычный подход при организации вычислений для многопроцессорных вычислительных систем с общей памятью – создание новых параллельных методов на основе обычных последовательных программ, в которых или автоматически компилятором, или непосредственно программистом выделяются участки независимых друг от друга вычислений. Возможности автоматического анализа программ для порождения параллельных вычислений достаточно ограничены, и второй подход является преобладающим. При этом для разработки параллельных программ могут применяться как новые алгоритмические языки, ориентированные на параллельное программирование, так и уже имеющиеся языки, расширенные некоторым набором операторов для параллельных вычислений.

Широко используемый подход состоит и в применении тех или иных библиотек, обеспечивающих определенный программный интерфейс (application programming interface, API) для разработки параллельных программ. В рамках такого подхода наиболее известны Windows Thread API. Однако первый способ применим только для ОС семейства Microsoft Windows, а второй вариант API является достаточно трудоемким для использования и имеет низкоуровневый характер [3].

1.4 Параллельный алгоритм Винограда

Трудоемкость алгоритма Винограда имеет сложность $O(nmk)$ для умножения матриц $n1 \times m1$ на $n2 \times m2$. Чтобы улучшить алгоритм, следует распараллелить ту часть алгоритма, которая содержит 3 вложенных цикла. Помимо этого, можно объединить всю остальную часть алгоритма и ее тоже распараллелить, тем самым разбив алгоритм на два последовательно выполняющихся участка.

В результирующей матрице каждая ячейка вычисляется независимо от других, поэтому, вычисляя отдельные строки разными потоками, получится не сталкиваться с проблемой "разделяемых данных"между потоками, так как каждый поток будет отвечать за свой участок итоговой матрицы.

1.5 Вывод

В данном разделе был рассмотрен алгоритм Винограда и способ его распараллеливания.

2 Конструкторская часть

В данной части будут рассмотрены схема алгоритма винограда и модель его распараллеливания.

2.1 Схема алгоритма Винограда

На рисунке 1 приведена схема алгоритма Винограда.

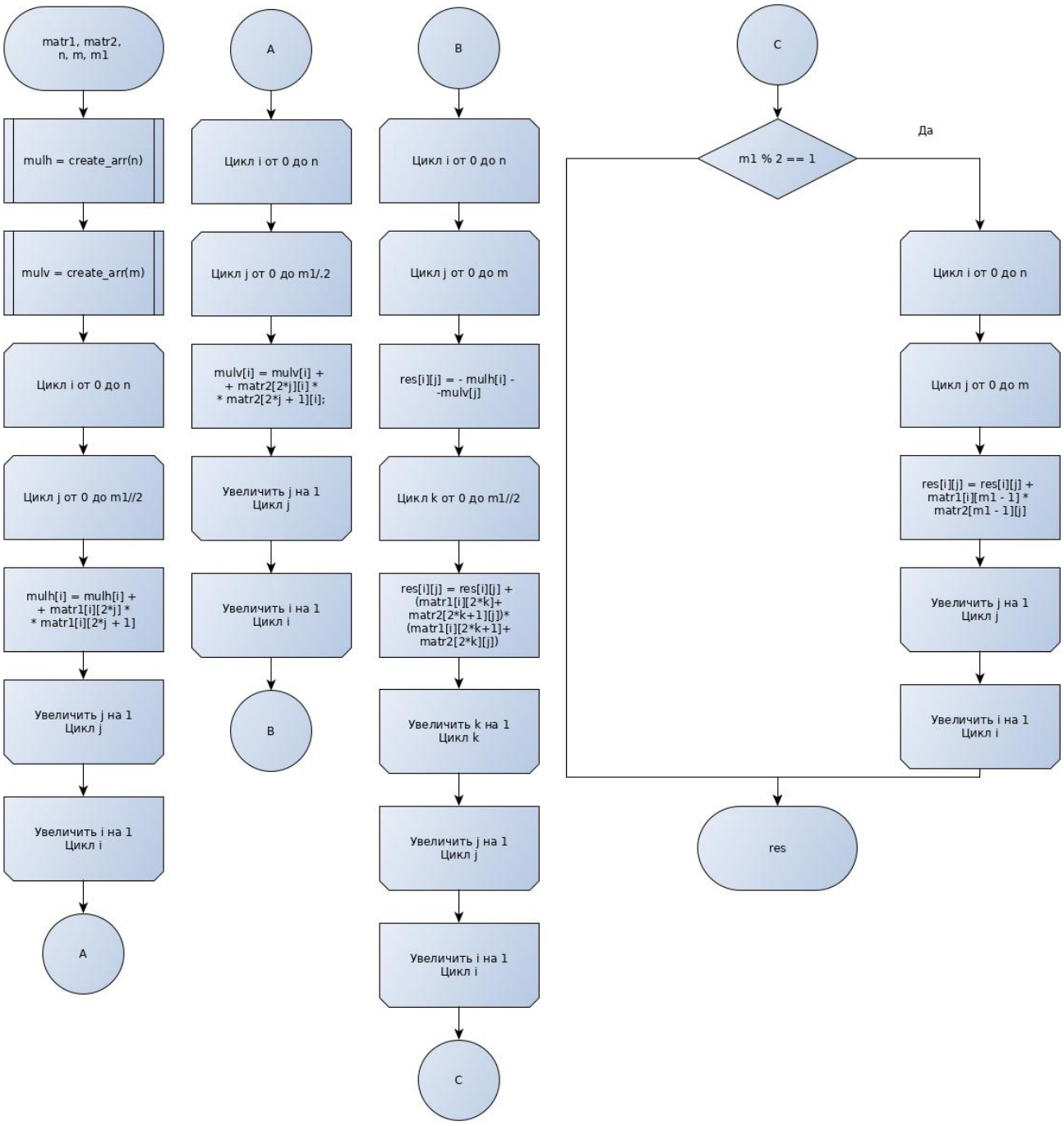

Рис. 1: Алгоритм Винограда

2.2 Модель организации параллельных вычислений

Алгоритм Винограда можно разбить условно на 4 части:

1. вычисление вектора mulh(на Рис. 1 эта часть находится в промежутке от начала алгоритма до соеденителя A);
2. вычисление вектора mulv (на Рис. 1 от A до B);
3. основная часть(на Рис. 1 от A до C);
4. дополнительные вычисления в случае нечетного количества столбцов в первой матрице(на Рис. 1 от C до конца алгоритма).

   В таком случае можно распараллелить часть 3, модель представлена на рисунке (2), квадраты на ней - этапы алгоритма, перегородки - ожидание всех потоков:

Рис. 2: Алгоритм Винограда

2.3 Вывод

В данном разделе были рссмотрены схемы алгоритмов, таких как: стандартный алгоритм умножения матриц, алгоритм Винограда и оптимизированный алгоритм Винограда.

3 Технологическая часть

В данном разделе будут приведены листинги алгоритмов на языке c, оптимизации для алгоритма Винограда, оценена трудоемкости каждого алгоритма.

3.1 Требования к программному обеспечению

Входные данные - матрица1, матрица2, их размеры. Выходные данные - произведение матриц.

Рис. 3: IDEF0-диаграмма, описывающая алгоритм умножения матриц

3.2 Средства реализации

Программа была написана на языке C. Проект выполнен с помощью сборки в утилите make. Программа корректно работает с пустыми и неправильно введенными данными.

3.3 Листинг кода

В листингах 2-4 приведены все рассматриваемые в рамках данной лабораторной работы алгоритмы, написанные на языке C.

Листинг 1: Стандартный алгоритм

Листинг 2: Алгоритм Винограда

Листинг 3: Оптимизированный алгоритм Винограда

3.4 Вывод

4 Экспериментальная часть

4.1 Примеры работы программы

4.2 Постановка эксперимента

4.3 Результаты эксперимента

4.4 Вывод

В данном разделе алгоритмы были рассмотрены на предмет правильности работы, что было показано на примерах из листингов 7-9. Все алгоритмы оказались верны. Также был произведен анализ по затрачиваемому процессорному времени на каждый из алгоритмов, из которого было выявлено, что алгоритм Винограда начинает работать быстрее в худшем случае при размерности, превышающую 201x201.

Заключение

В ходе лабораторной работы были исследованы алгоритмы умножения матриц: стандартный, Винограда, и оптимизированный алгоритм Винограда. Для каждого алгоритма была посчитана трудоемкость в выбранной модели вычислений. Помимо этого, экспериментально были произведены замеры времени работы каждого из рассматриваемых алгоритмов.

Список литературы

[1] И. В. Белоусов(2006), Матрицы и определители, учебное пособие по линейной алгебре, с. 1 - 16

[2] Дж. Макконнелл. Анализ алгоритмов. Активный обучающий подход.-М.:Техносфера, 2009.

[3] Константин Баркалов, Владимир Воеводин, Виктор Гергель. Intel Parallel Programming [Электронный ресурс], - режим доступа https://www.intuit.ru/studies/courses/4447/983/lecture/14925

[4] Pthreads: Потоки в русле POSIX. [Электронный ресурс], - режим доступа https://habr.com/ru/post/326138/

[5] Multithreading in C++ [Электронный ресурс], - режим доступа https://www.geeksforgeeks.org/multithreading-in-cpp/