

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Нижегородский государственный университет им. Н.И.
Лобачевского»
Национальный исследовательский университет

Институт информационных технологий, математики и механики

ЛАБОРАТОРНАЯ РАБОТА
по параллельному программированию
“Умножение плотных матриц. Алгоритм Штрассена.”

Выполнил:

студент группы 381506-3
Новоселова Екатерина
Андреевна

_____ Подпись

Проверил:

Доцент кафедры МОСТ
Кандидат технических наук
Сысоев Александр
Владимирович

_____ Подпись

Нижний Новгород
2018

1. Постановка задачи

Требуется освоить метод Штрассена для умножения квадратных матриц, запрограммировать алгоритм решения задачи на языке C++ с использованием OpenMP и библиотека Intel Threading Building Blocks (TBB).

2. Алгоритм Штрассена

Пусть A и B — две $(n \times n)$ -матрицы, причём n — степень числа 2. Тогда можно разбить каждую матрицу A и B на четыре $((n/2) \times (n/2))$ -матрицы и через них выразить произведение матриц A и B :

Определим новые элементы:

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}, \quad B = \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix}, \quad C = \begin{bmatrix} C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{bmatrix}$$

Таким образом, нам нужно всего 7 умножений на каждом этапе рекурсии. Элементы матрицы C выражаются из P_k по формулам:

$$P_1 := (A_{1,1} + A_{2,2})(B_{1,1} + B_{2,2})$$

$$P_2 := (A_{2,1} + A_{2,2})B_{1,1}$$

$$P_3 := A_{1,1}(B_{1,2} - B_{2,2})$$

$$P_4 := A_{2,2}(B_{2,1} - B_{1,1})$$

$$P_5 := (A_{1,1} + A_{1,2})B_{2,2}$$

$$P_6 := (A_{2,1} - A_{1,1})(B_{1,1} + B_{1,2})$$

$$P_7 := (A_{1,2} - A_{2,2})(B_{2,1} + B_{2,2})$$

Рекурсивный процесс продолжается n раз, до тех пор пока размер матриц $C_{i,j}$ не станет достаточно малым, далее используют обычный метод умножения матриц. Это делают из-за того, что алгоритм Штрассена теряет эффективность по сравнению с обычным на малых матрицах в силу большего числа сложений.

$$C_{1,1} = P_1 + P_4 - P_5 + P_7$$

$$C_{1,2} = P_3 + P_5$$

$$C_{2,1} = P_2 + P_4$$

$$C_{2,2} = P_1 - P_2 + P_3 + P_6$$

3. Реализации и схема распараллеливания

Для реализации алгоритма Штрассена нам понадобятся дополнительные функции. Как было сказано выше, алгоритм работает только с квадратными матрицами, размерность которых равна степени 2, поэтому для корректной работы алгоритма нужно увеличить размер матриц до ближайшей степени 2 и заполнить расширенную часть нулями.

Для распараллеливания алгоритма был применен механизм задач (task). Для каждого рекурсивного вызова создается отдельная задача.

4. Подтверждение корректности. Результаты экспериментов по оценке масштабируемости

Алгоритм Штрассена дает преимущество при малом количестве потоков. Во-первых, он требует большое количество дополнительной памяти (особенно при распараллеливании), что ограничивает максимальный размер перемножаемых матриц. Во-вторых, рекурсивная структура алгоритма хуже поддается распараллеливанию, так как схема распараллеливания получается многоуровневой с синхронизацией между этими уровнями.

