

Отчет о вычислительных экспериментах по перемножению матриц

Сотникова Виктория, 307 группа

1 Введение

В данном отчете представлены результаты вычислительных экспериментов по перемножению матриц с использованием наивного алгоритма. Исследовано влияние размеров матриц и количества потоков OpenMP на время выполнения алгоритма.

2 Алгоритм перемножения матриц

Каждый элемент матрицы C вычисляется по следующей формуле:

$$C_{ij} = \sum_{k=1}^N A_{ik} \cdot B_{kj}$$

Вычислительная сложность данного алгоритма определяется следующим образом:

Для вычисления одного элемента C_{ij} требуется выполнить N операций умножения (по одному для каждого элемента строки матрицы A и соответствующего элемента столбца матрицы B). Также требуется $N - 1$ операции сложения для суммирования этих произведений.

Результирующая матрица C имеет размерность $M \times K$, что означает, что необходимо вычислить $M \times K$ элементов.

Таким образом, общее количество операций умножения составляет $M \times K \times N$, что приводит к общей вычислительной сложности $O(MNK)$

Где M — количество строк в матрице A , N — количество столбцов в матрице A и строк в матрице B , K — количество столбцов в матрице B .

3 ТТХ Вычислительной системы

Для проведения экспериментов использовалась система с следующими характеристиками:

- Модель ЦП: 12th Gen Intel(R) Core(TM) i5-12500H
- Количество ядер: 12
- Базовая тактовая частота: 2.50 GHz
- Объем оперативной памяти: 16 GiB

4 Результаты экспериментов

Эксперименты проводились с различными размерами матриц и количеством потоков OpenMP. Результаты представлены в таблице 1.

Таблица 1: Результаты вычислительных экспериментов

Эксперимент	M	N	K	P	Время T (с)	Время T (с)
					(непараллельное)	(параллельное)
1	1000	2000	1000	1	8.64854	9.33026
2	1000	2000	1000	2	8.68694	4.83309
3	1000	2000	1000	4	8.99503	2.55622
4	1000	2000	1000	8	9.06517	1.86212
5	1000	2000	1000	16	9.32597	1.59962
6	1000	1000	1000	1	4.36037	4.53064
7	2000	1000	1000	2	8.58866	4.63927
8	4000	1000	1000	4	17.4129	4.82163
9	8000	1000	1000	8	35.1591	7.71922
10	16000	1000	1000	16	71.1889	11.8814

5 Анализ результатов

Время выполнения (Duration): с увеличением количества потоков P время выполнения параллельной версии уменьшается, что ожидаемо, так как задача распределяется между несколькими потоками. Однако, Duration Parallel не всегда меньше Duration Serial, что указывает на наличие накладных расходов на управление потоками и синхронизацию.

Ускорение (Speedup): отношение времени выполнения последовательной версии к времени выполнения параллельной версии. $\text{Speedup} > 1$ говорит о том, что параллельная версия выполняется быстрее, чем последовательная. В данном случае Speedup увеличивается с увеличением P до 4, после чего начинает снижаться. Это указывает на то, что при более высоком количестве потоков накладные расходы начинают преобладать над выигрышем от параллелизма.

Эффективность (Efficiency): отношение Speedup к количеству потоков ($E = S/P$). Efficiency показывает, насколько эффективно используются потоки. Значение $E < 1$ говорит о том, что не все потоки работают эффективно. В данном случае эффективность снижается с увеличением P , что указывает на то, что добавление дополнительных потоков не приводит к пропорциональному увеличению производительности.

Стоимость (Cost): стоимость вычислений C увеличивается с увеличением P . Это связано с тем, что при большем количестве потоков требуется больше ресурсов и времени для управления ими.

Суммарные накладные расходы (Total Overhead TO): накладные расходы (TO) значительно возрастают с увеличением количества потоков. Это указывает на то, что управление большим количеством потоков требует больше времени и ресурсов.

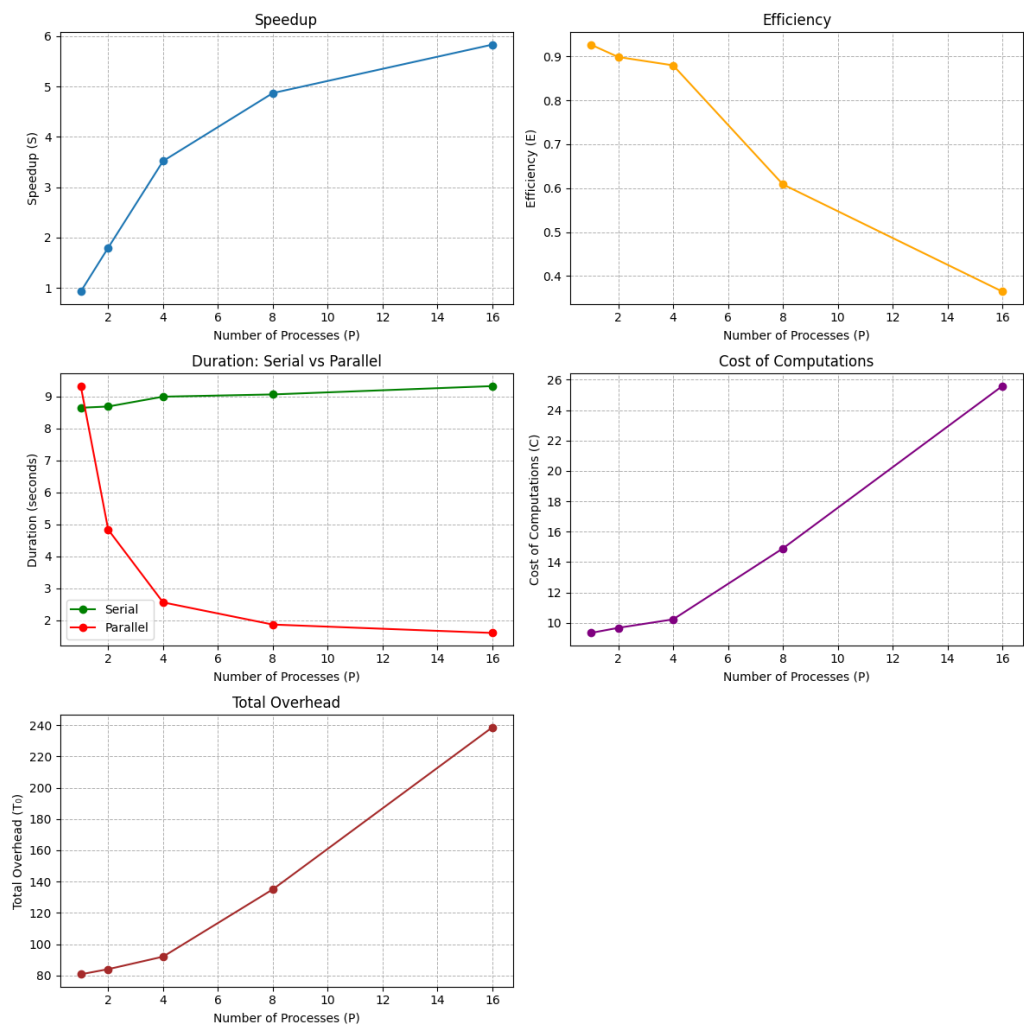


Рис. 1: График производительности метрик параллельных вычислений

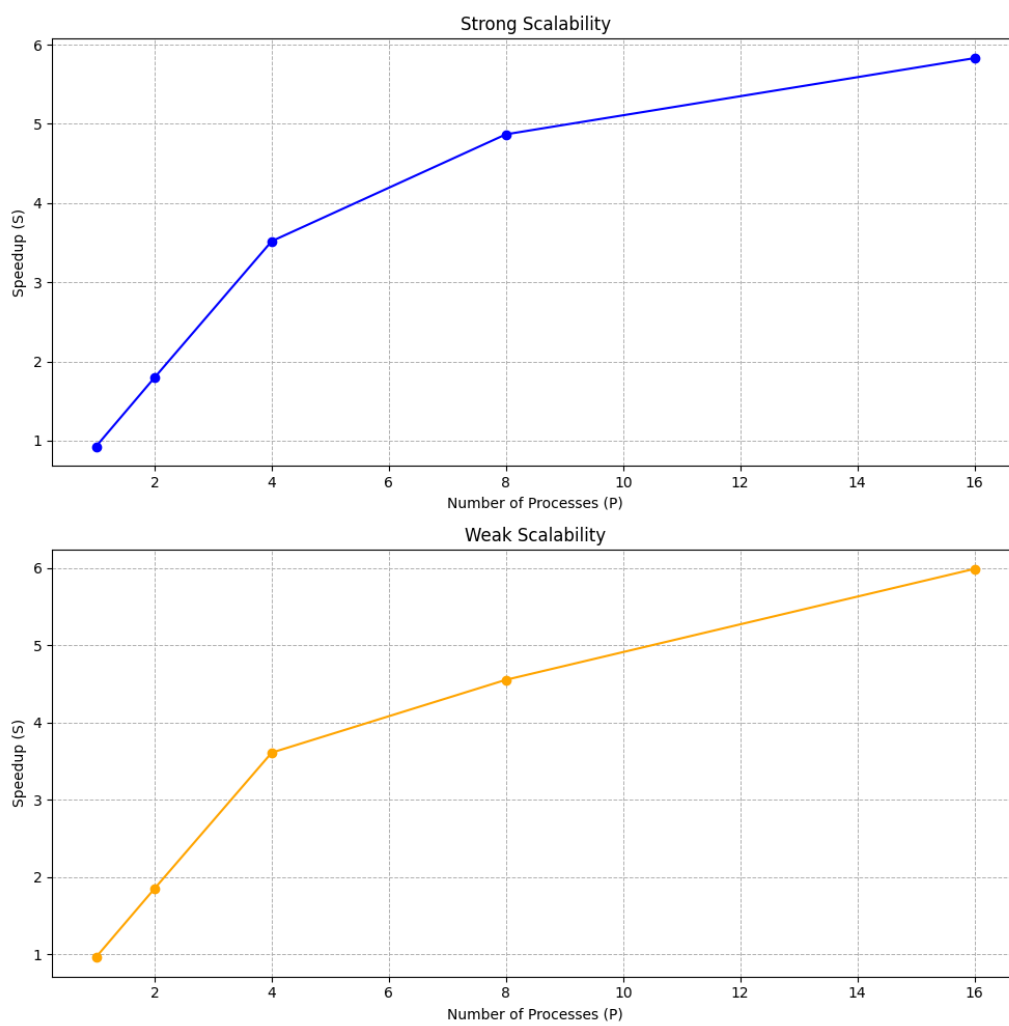


Рис. 2: Графики сильной и слабой масштабируемости

6 Заключение

Оптимальное количество потоков: На основе полученных данных оптимальное количество потоков для достижения максимального ускорения составляет около 4-8. При дальнейшем увеличении числа потоков наблюдается снижение эффективности и увеличение накладных расходов.

Переход к параллелизму: Параллелизация не всегда приводит к линейному улучшению производительности из-за накладных расходов на управление потоками и возможных конфликтов между ними.

Размерности в задаче: увеличение размерностей в задаче приводит к улучшению производительности параллельного выполнения, но также вызывает увеличение накладных расходов и снижение эффективности при высоких значениях количества потоков P .

Исходный код доступен в репозитории.