1. Перемножение матриц алгоритмом Фокса

Постановка задачи

1) Реализовать последовательный алгоритм перемножения матриц.

2) Реализовать программу блочного умножения матриц (Алгоритм Фокса), используя технологию MPI.

3) Провести ряд тестов. Сравнить ускорение параллельного и не параллельного алгоритма.

Последовательный алгоритм перемножения матриц

Произведение матрицы A размерами m*n и матрицы B размерами n*k является матрица C, элементы которой вычисляются по формуле:

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \ldots + a_{in}b_{nj} = \sum_{s=1}^{n} a_{is}b_{sj},$$

Алгоритм Фокса перемножения матриц

Используется блочная схема разбиения матрицы. При таком способе разделения данных исходные матрицы A, B и результирующая матрица C представляются в виде наборов блоков. Далее предполагается что все матрицы являются квадратными размера n×n, количество блоков по горизонтали и вертикали одинаково и равно q (т.е. размер всех блоков равен k×k, k=n/q). При таком представлении данных операция матричного умножения матриц A и B в блочном виде может быть представлена так:

$$\begin{pmatrix} A_{00}A_{01}\ldots A_{0q-1} \\ \ldots \\ A_{q-10}A_{q-11}\ldots A_{q-1q-1} \end{pmatrix} \times \begin{pmatrix} B_{00}B_{01}\ldots B_{0q-1} \\ \ldots \\ B_{q-10}B_{q-11}\ldots B_{q-1q-1} \end{pmatrix} = \begin{pmatrix} C_{00}C_{01}\ldots C_{0q-1} \\ \ldots \\ C_{q-10}C_{q-11}\ldots C_{q-1q-1} \end{pmatrix},$$

,где каждый блок результирующей матрицы C определяется по формуле $C_{ij} = \sum_{s=0}^{q-1} A_{is}B_{sj}.$

За основу параллельных вычислений для матричного умножения при блочном разделении данных принят подход, при котором базовые подзадачи отвечают за вычисления отдельных блоков матрицы C и при этом в подзадачах на каждой итерации расчетов располагается только по одному блоку исходных матриц A и B. Для нумерации подзадач будем использовать индексы размещаемых в подзадачах блоков матрицы C, т.е. подзадача (i,j) отвечает за вычисление блока Cij – тем самым, набор подзадач образует квадратную решетку, соответствующую структуре блочного представления матрицы C.

В соответствии с алгоритмом Фокса в ходе вычислений на каждой базовой подзадаче (i,j) располагается четыре матричных блока:

1. Блок Cij матрицы C, вычисляемый подзадачей;

2. Блок Aij матрицы A, размещаемый в подзадаче перед началом вычислений;

3. Блоки A'ij , B'ij матриц A и B, получаемые подзадачей в ходе выполнения вычислений.

Выполнение параллельного метода включает:

1. Этап инициализации, на котором каждой подзадаче (i,j) передаются блоки Aij, Bij и обнуляются блоки Cij на всех подзадачах;

2. Этап вычислений, в рамках которого на каждой итерации l, l от нуля (включительно) до q, осуществляются следующие операции:

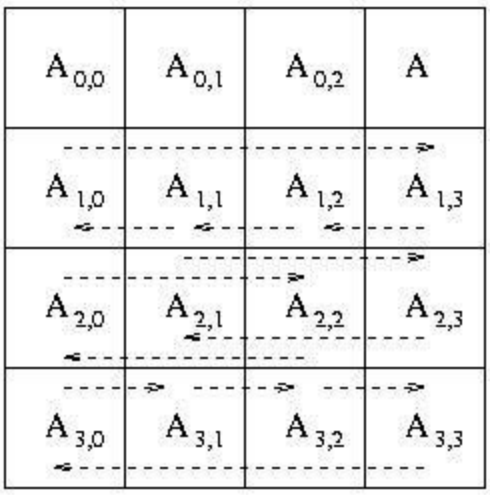
a. Для каждой строки i (i от нуля включительно и строго до q) блок Aij подзадачи (i,j) пересылается на все подзадачи той же строки i решетки; индекс j, определяющий положение подзадачи в строке, вычисляется в соответствии с выражением j=(i+l)mod q, где mod есть операция получения остатка от целочисленного деления;

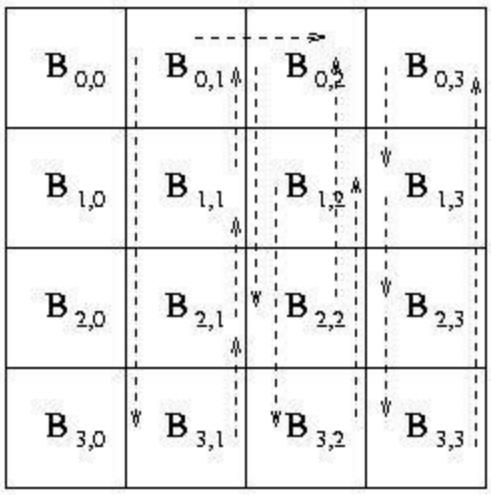
b. Полученные в результаты пересылок блоки A'ij, B'ij каждой подзадачи (i,j) перемножаются и прибавляются к блоку Cij

c. Блоки B'ij каждой подзадачи (i,j) пересылаются подзадачам, являющимся соседями сверху в столбцах решетки подзадач (блоки подзадач из первой строки решетки пересылаются подзадачам последней строки решетки).

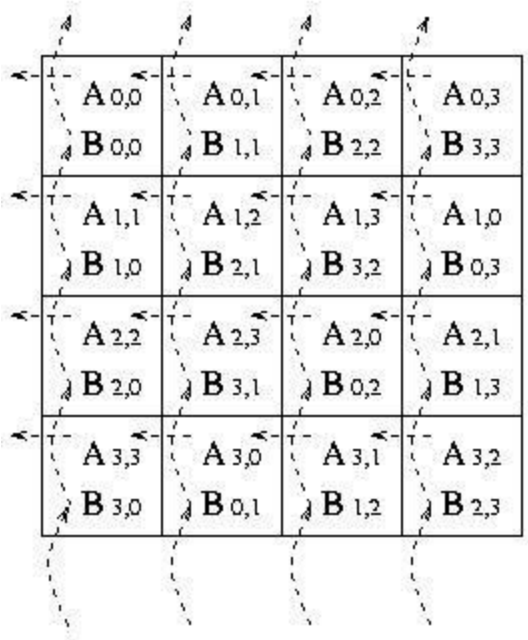
Демонстрация алгоритма

Изначально блоки матриц-операндов A и B располагаются следующим образом:

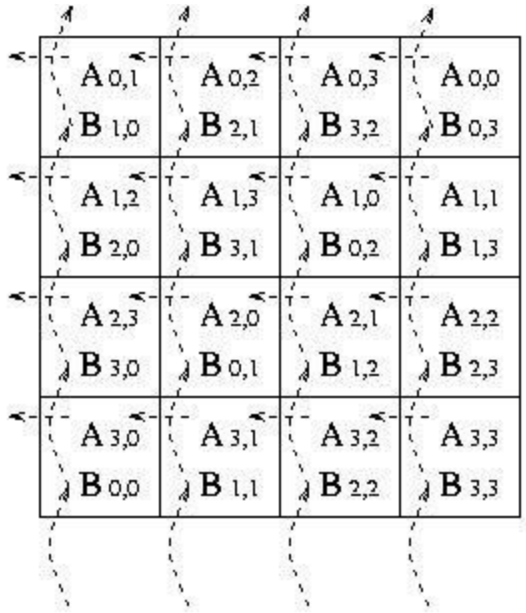




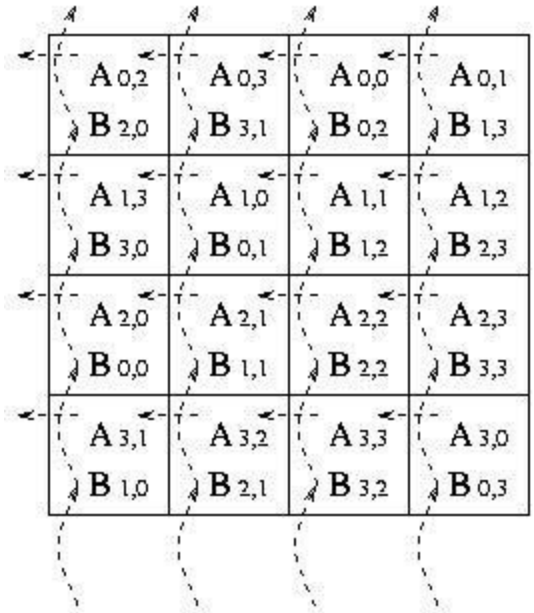
После первой итерации вычислений перемножены были следующие блоки матриц A и B:



После первого сдвига перемножены будут следующие блоки матриц A и B:



После второго сдвига перемножены будут следующие блоки матриц A и B:



После третьего сдвига перемножены будут следующие блоки матриц A и B:

A 0,3 B 3,0	A 0,0 B 0,1	A 0,1 B 1,2	A 0,2 B 2,3
A 1,0 B 0,0	A 1,1 B 1,1	A 1,2 B 2,2	A 1,3 B 3,3
A 2,1 B 1,0	A 2,2 B 2,1	A 2,3 B 3,2	A 2,0 B 0,3
A 3,2 B 2,0	A 3,3 B 3,1	A 3,0 B 0,2	A 3,1 B 1,3

Реализация с использованием MPI.

Основные этапы разработки параллельного алгоритма:

1. Создание виртуальной декартовой топологии

2. Определение размеров объектов и ввод исходных данных

3. Завершение процесса вычислений

4. Распределение данных между процессами

5. Начало реализации параллельного алгоритма матричного умножения

6. Рассылка блоков матрицы A

7. Циклический сдвиг блоков матрицы B вдоль столбцов процессорной решетки

8. Умножение матричных блоков

9. Сбор результатов