

15 СТОЛБЦОВО-ОРИЕНТИРОВАННЫЕ АЛГОРИМЫ УМНОЖЕНИЯ МАТРИЦ НА КОЛЬЦЕ (СИСТОЛИЧЕСКИЙ АЛГОРИТМ НА ПРОЦЕССОРНОМ КОЛЬЦЕ)

Equation Chapter (Next) Section 1 Хотим посчитать

$$\begin{aligned} D &= C + AB \\ A, B, C, D &\in \mathbb{R}^{n \times n} \end{aligned} \quad (1.1)$$

параллельно, то есть есть p процессоров, а ранг текущего — μ . На каждый процессор без ограничения общности выделим $r = \frac{n}{p}$ столбцов.

При этом всем, матрица у нас будет храниться по блочным столбцам, то есть

$$\begin{aligned} \left(D_1 \mid D_2 \mid \dots \mid D_p \right) &= \left(C_1 \mid C_2 \mid \dots \mid C_p \right) + \\ &+ \left(A_1 \mid A_2 \mid \dots \mid A_p \right) \left(B_1 \mid B_2 \mid \dots \mid B_p \right) \end{aligned} \quad (1.2)$$

Каждый процессор будет в итоге считать свой блочный столбец, то есть

$$D_\mu = C_\mu + AB_\mu = C_\mu + \sum_{k=1}^p A_k B_{k\mu} \quad (1.3)$$

Видно, что каждому процессору понадобится каждый блочный столбец A_k . Ими они и будут обмениваться этаким хороводиком. Приведем

код алгоритма:

```
// инициализация
// индексы нашего столбца
col = (mu - 1) * r + 1 : mu * r
// блочные столбцы, которыми мы будем пользоваться
// результат положим в матрицу C
A_loc = A(:, col)
B_loc = B(:, col)
C_loc = C(:, col)
// индексы соседей
left = (mu - 2 + p) % p + 1
right = mu % p + 1
```

```
// вычисления
for t = 1 : p
    send(A_loc, right)
    recv(A_loc, left)
    // индекс блочного столбца, который сейчас лежит в A_loc
    tau = (mu - t - 1 + p) % p + 1
    // соответствующие ему строчные индексы
    row_tau = (tau - 1) * r + 1 : tau * r
    // изменяем ответ
    C_loc = C_loc + A_loc * B_loc(row_tau, :)
end
```