



矩阵乘积并行算法 (OpenMP)

矩阵矩阵并行乘积

- 串行算法
- 自动并行
- 手工并行

矩阵乘积串行算法

$$C = AB$$

$$A \in \mathbb{R}^{M \times L}, B \in \mathbb{R}^{L \times N}$$

■ 串行算法

▶ IJK 顺序

```
for(i=0; i<M; i++)  
  for(j=0; j<N; j++)  
    for(k=0; k<L; k++)  
      C[i][j]=C[i][j] + A[i][k]*B[k][j];
```

▶ IKJ 顺序、KIJ 顺序、JIK 顺序、KJI 顺序、JKI 顺序

▶ 快速算法：Strassen 算法等

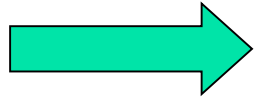
示例程序：hw01

并行算法：自动并行

$$C = AB$$

$$A \in \mathbb{R}^{M \times L}, B \in \mathbb{R}^{L \times N}$$

自动并行



循环共享结构

```
#pragma omp parallel for shared(A,B,C) .....  
for(i=0; i<M; i++)  
  for(j=0; j<N; j++)  
  {  
    C[i][j]=0;  
    for(k=0; k<L; k++)  
      C[i][j]=C[i][j] + A[i][k]*B[k][j];  
  }
```

OMP_matmul_for.c

并行：手工并行

$$AB = \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_{p-1} \end{bmatrix} [B_0 \ B_1 \ \cdots \ B_{p-1}] = \begin{bmatrix} A_0B_0 & A_0B_1 & \cdots & A_0B_{p-1} \\ A_1B_0 & A_1B_1 & \cdots & A_1B_{p-1} \\ \vdots & & \ddots & \\ A_{p-1}B_0 & A_{p-1}B_1 & \cdots & A_{p-1}B_{p-1} \end{bmatrix}$$

手工并行

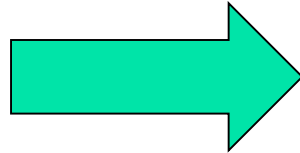
任务分配：由用户分配计算任务
(即每个线程负责计算 C 的哪些部分)

C ：按行分块、按列分块、二维分块

假定： M, L, N 均能被 p 整除，其中 p 为线程个数

并行：按行分配任务

记 $A = \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_{p-1} \end{bmatrix}$



$$C = AB = \begin{bmatrix} A_0 B \\ A_1 B \\ \vdots \\ A_{p-1} B \end{bmatrix}$$

按行分配任务

- 第 i 号线程负责计算 C_i ，其中 $C_i = A_i B$
- 实际计算时，对 A 的划分可以采用**连续**方式或**卷帘**方式

举例

例：按行分配任务，并行计算矩阵乘积，其中

$$A = [a_{ij}] \in \mathbb{R}^{n \times n}, \quad a_{ij} = \frac{1}{i + j - 1}$$

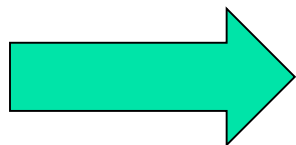
$$B = [b_{ij}] \in \mathbb{R}^{n \times n}, \quad b_{ij} = i + j - 1$$

(取 $n=1024, p=4$)

[OMP_MatMul_rc.c](#)

并行：按列分配任务

$$B = [B_0, B_1, \dots, B_{q-1}]$$



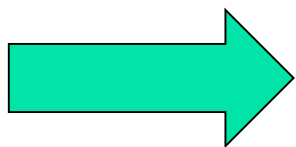
$$C = [AB_0, AB_1, \dots, AB_{q-1}]$$

按列分配任务

- 第 i 号线程负责计算 C_i ，其中 $C_i = Ab_i$
- 实际计算时，对 B 的划分可以采用**连续**方式或**卷帘**方式

并行：二维划分

$$A = \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_{p-1} \end{bmatrix}, \quad B = [B_0, B_1, \dots, B_{q-1}]$$



$$C = \begin{bmatrix} A_0 B_0 & A_0 B_1 & \cdots & A_0 B_{q-1} \\ A_1 B_0 & A_1 B_1 & \cdots & A_1 B_{q-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p-1} B_0 & A_{p-1} B_1 & \cdots & A_{p-1} B_{q-1} \end{bmatrix}$$

行列划分

- 线程总数为 $nthreads = p \times q$ ，其中 p 和 q 为两个正整数
- 第 (i, j) 号线程负责计算 C_{ij} ，其中 $C_{ij} = A_i B_j$
- 实际计算时，也可以采用卷帘方式分配任务（即划分矩阵 A 和 B ）