并行计算 HW5

PB18111697 王章瀚

2021 年 5 月 25 日

9.S1

试将 Cannon 分块乘法算法 9.5 改为共享存储 PRAM-EREW 模型上的算法, 并分析其时间复杂度.

修改后如下:

Algorithm 1 CANNON

for all $P_{i,j}$ **par- do**
  $C_{i,j} \leftarrow 0$
  for $k = 0$ to $\sqrt{p} - 1$ **do**
    $C_{i,j} \leftarrow A_{i,(i+j+k) mod \sqrt{p}} \cdot B_{(i+j+k) mod \sqrt{p}, j}$

复杂度包括 $\sqrt{p}$ 次迭代, 每次是一个矩阵乘法 $(\frac{n}{\sqrt{p}})^3$, 因此总的为 $\Theta(\sqrt{p} \cdot (\frac{n}{\sqrt{p}})^3) = \frac{n^3}{p}$

9.9

9.9 **算法 9.7 给出了 n^2 个处理器的并行系统上用 PRAM-CREW 模型施行两个 $n \times n$ 矩阵相乘的算法。假定存储器的读写时间为 t_a，两个元素的乘-加时间为 t_c。试分析该算法的并行运行时间。**

算法 9.7 PRAM-CREW 上矩阵相乘算法

输入:$A_{n\times n}$, $B_{n\times n}$
输出:$C_{n\times n}$
Begin
  (1) 将 n^2 个处理器组织成 $n \times n$ 的网孔
  (2) **for** each $P_{i,j}$ **do**
    (2.1) $c_{i,j} = 0$
    (2.2) **for** $k=0$ **to** $n-1$ **do**
        $c_{i,j} = c_{i,j} + a_{i,k} \times b_{k,j}$
      endfor
   endfor
End

每个进程在一个 k 的迭代轮次中, 需要收集到来自同行第 k 列和同列第 k 行的数据, 这个运行时间是 t_a. 而乘加运算是 t_c, 因此一个迭代需要 $(t_a + t_c)$, 所以该算法的并行运行时间为 $n(t_a + t_c)$.(忽略对 $c_{i,j}$赋初值时间)