Homework 3

PB17000297 罗晏宸

April 13 2020

1 Algorithm 10.1

以下是上三角方程组回代解法的串行算法的形式化描述。

算法 1 SISD 上回代求解上三角方程组算法

输入： $\boldsymbol{A}_{n\times n}, \boldsymbol{b} = (b_1, \cdots, b_n)^\top$

输出： $\boldsymbol{x} = (x_1, \cdots, x_n)^\top$

1: **Begin**
2:     **for** $i = n$ **downto** 1 **do**
3:         $x_i = b_i / a_{ii}$
4:         **for** $j = 1$ **to** $i - 1$ **do**
5:             $b_j = b_j - a_{ji} x_i$
6:             $a_{ji} = 0$
7:         **end for**
8:     **end for**
9: **End**

1 请指出串行算法哪些部分可以并行化。

2 写出并行算法的形式化描述（需要注明计算模型类型），分析你的算法的时间复杂度

解

1 算法第4行的 j 循环可以并行化。

2 在 UMA 模型上回代求解上三角方程组的并行算法如算法2所示。由2和4两行循环可知复杂度为 $p(n)=p$，$t(n)=O(n)$

算法 2 UMA 上回代求解上三角方程组算法

输入： $\boldsymbol{A}_{n\times n},\boldsymbol{b}=(b_1,\cdots,b_n)^\mathsf{T}$

输出： $\boldsymbol{x}=(x_1,\cdots,x_n)^\mathsf{T}$

```
1:  Begin
2:      for i = n downto 1 do
3:          x_i = b_i / a_ii
4:          for all P_j, 1 ⩽ j ⩽ p do
5:              for k = j to i − 1 step p do        ▷ p 是处理器数
6:                  b_k = b_k − a_ki x_i
7:                  a_ki = 0
8:              end for
9:          end for
10:     end for
11: End
```

---

2 Exercise 7.10

顶点倒塌法是非常有名的求图的连通分量的算法，其基本思想是：连通的相邻的顶点可以合并成一个超顶点，并以它们中最小标号者标记之；此过程可继续在已合并的超顶点之间进行。在下列的算法中。$C(i)$ 表示与 i 相邻的最小的超顶点号码；$D(i)$ 表示顶点 i 所属连通分量的最小标号的顶点；$C(i)=\min_j\{D(j)|_{A(i,j)=1,D(i)\neq D(j)}\}$ 语句为每个顶点 i 找与它不属于相同分量的相邻的最小号码的顶点 j；语句 $C(i)=\min_j\{C(j)|_{D(j)=i,C(j)\neq i}\}$ 表示把每个超顶点的根连到最小号码的相邻的超顶点的根上。Hirschberg 的求连通分量算法如下。

算法 3 PRAM-CREW 上 Hirschberg 求连通分量算法

输入： 邻接矩阵 $\boldsymbol{A}_{n\times n}$

输出： 向量 $\boldsymbol{D}(0:n-1)$，其中 $D(i)$ 表示向量 $\boldsymbol{D}$ 的分量

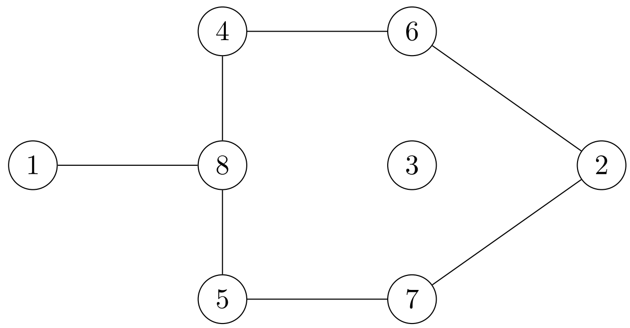
```
Begin
    for all i : 0 ⩽ i ⩽ n − 1 par-do        ▷ 初始化
        D(i) = i
```

end for
for $\lceil \log n \rceil$ iterations **do**
for all $i, j : 0 \leqslant i, j \leqslant n-1$ **par-do** ▷ 找相邻的最小者
$C(i) = \min\limits_{j} \{D(j)|_{A(i,j)=1, D(i) \neq D(j)}\}$
if none **then**
$C(i) = D(i)$
end if
end for
for all $i, j : 0 \leqslant i, j \leqslant n-1$ **par-do** ▷ 找每个超顶点的最小相邻超顶点
$C(i) = \min\limits_{j} \{C(j)|_{D(j)=i, C(j) \neq i}\}$
if none **then**
$C(i) = D(i)$
end if
end for
for all $i : 0 \leqslant i \leqslant n-1$ **par-do**
$D(i) = C(i)$
end for
for $\lceil \log n \rceil$ iterations **do**
for all $i : 0 \leqslant i \leqslant n-1$ **par-do**
$C(i) = C(C(i))$
end for
end for
for all $i : 0 \leqslant i \leqslant n-1$ **par-do**
$D(i) = \min\{C(i), D(C(i))\}$
end for
end for
End

(1) 试分析算法3的复杂度 $t(n)$ 和 $p(n)$。

(2) 给定如图1所示无向图，试用算法3逐步求出该图的连通分量

解



顶点	1	2	3	4	5	6	7	8
分量	1	2	3	4	5	6	7	8

图 1: 待求连通分量的无向图

(1) 注意到第5行最外侧的循环迭代 $\lceil \log n \rceil$ 次，在循环内第21行循环同样迭代 $\lceil \log n \rceil$ 次，同时外侧循环内的各 Do in parallel 循环语句指出需要的处理器数为 $n-1$，因此算法的复杂度为 $t(n) = (\log n)^2$，$p(n) = n$。

(2) 无向图共有 8 个结点，因此算法共进行 $\lceil \log 8 \rceil = 3$ 次迭代，迭代过程如图2所示。

顶点 i	1	2	3	4	5	6	7	8
$C(i)$								
$D(i)$	1	2	3	4	5	6	7	8

(a) 初始化

顶点 i	1	2	3	4	5	6	7	8
$C(i)$	1	2	3	4	5	2	2	1
$D(i)$	1	2	3	4	5	2	2	1

(b) 第一次迭代后的结果

顶点 i	1	2	3	4	5	6	7	8
$C(i)$	1	2	3	2	2	2	2	1
$D(i)$	1	2	3	2	2	2	2	1

(c) 第二次迭代后的结果

顶点 i	1	2	3	4	5	6	7	8
$C(i)$	1	1	3	1	1	1	1	1
$D(i)$	1	1	3	1	1	1	1	1

(d) 第三次迭代后的结果

图 2: 算法迭代过程