

并行计算 HW2

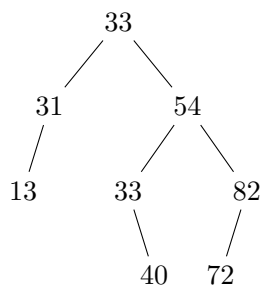
PB18111697 王章瀚

2021 年 4 月 4 日

1.

待排序序列是 (33, 31, 13, 54, 82, 33, 40, 72), 有 8 个处理器.

因为是示例, 不妨假设每次都是从左往右第一个处理器竞争到写入, 则有如下二叉树:



说明:

- 从 root 开始, 第一个是 33, 而后分为 [31, 13] 和 [54, 82, 33, 40, 72]
 - 左边 [31, 13] 中, 31 竞争到了, 于是先设置了 31, 然后自然是 13
 - 右边 [54, 82, 33, 40, 72] 中, 54 竞争到了, 于是先设置了 54, 分为了 [33, 40] 和 [82, 72]; 然后更进一步分别是 33 和 82

2.

二叉排序树并行地转化为有序数组, 写出算法, 时间复杂度, 并行计算模型.

不妨考虑类似顺序统计树的做法, 由此可以直接在 $O(\log n)$ 内得到每个节点的次序, 然后直接写入数组对应元素地址即可. 这里主要问题是统计每个节点对应的子树大小.

Algorithm 1 SORTED-TREE-TO-ARRAY

```
1: function SORTED-TREE-TO-ARRAY( $T$ )
2:   for  $i=1$  to  $n$  par- do                                     ▷ 初始化 size 数组 和 rank 数组,  $O(1)$ 
3:      $size[i] = 0$ 
4:      $rank[i] = -1$ 
5:   for  $i=1$  to  $n$  par- do                                       ▷ 初始化 parent 数组,  $O(1)$ 
6:     if  $LC_i$  available then
7:        $parent[LC_i] = i$ 
8:     if  $RC_i$  available then
9:        $parent[RC_i] = i$ 
10:  for  $k = 1$  to  $\lceil 1 + \log n \rceil$  do                             ▷ 初始化 level 数组, 自上而下遍历每层,  $O(\log n)$ 
11:    for proc  $i$  in level  $k$  par- do
12:       $level[i] = k$ 
13:  for  $k = \log p$  to  $1$  do                                       ▷ 逐层上传统计数量,  $O(\log n)$ 
14:    for proc  $i$  in level  $k$  par- do                               ▷ 初始化
15:       $size[i] = size[RC_i] + size[LC_i]$ 
16:
17:  for  $k = 1$  to  $\log p$  do                                         ▷ 逐层上传统计数量,  $O(\log n)$ 
18:    for proc  $i$  in level  $k$  par- do                               ▷ 初始化
19:      if  $i$  是右节点 then
20:         $rank[i] = rank[parent[i]] + 1$ 
21:      else if  $i$  是左节点 then
22:
```

根据前面的分析知道, 时间复杂度是 $O(\log n)$ 的, 所采用的并行计算模型是.