HW2

PB20000215 丁程

Ex6.3
给定序列（33，21，13，54，82，33，40，72）和8个处理器，试按照算法6.2构造一棵为在PRAM-CRCW模型上执行快排所用的二叉树
最终结果：
$root:33$
$processor1:f_1=1,LC_1=2,RC_1=4$
$processor2:f_2=1,LC_2=3,RC_2=2$
$processor3:f_3=2,LC_3=3,RC_3=3$
$processor4:f_4=1,LC_4=6,RC_4=5$
$processor5:f_5=4,LC_5=8,RC_5=5$
$processor6:f_6=4,LC_6=7,RC_6=6$
$processor7:f_7=6,LC_7=7,RC_7=7$
$processor8:f_8=5,LC_8=8,RC_8=8$
二叉树如下：

```
        33
       /  \
     21    54
    /     /  \
  13     33    82
        /    /
      40    72
```

2.如何实现算法6.3所构造二叉排序树并行地转化为有序数组，写出实现的并行算法，其时间复杂度和并行计算模型各是什么？
使用并行中序遍历算法
用CRCW模型，

```
Begin
    let P[0...n-1] be a new array of size n
    let S[0...p-1] be a new array of size p
    for each processor i do
        P[i] = fi
        S[i] = 1
        barrier

        for j = 1 to log n do
            if i < 2^(j-1) then
                if P[S[i]] < S[i] then
                    S[i] = LC[S[i]]
                else
                    S[i] = RC[S[i]]
                end if
            end if
            barrier

            if i >= 2^(j-1) and P[S[i]] >= 0 then
                if P[S[i]] < S[P[S[i]]] then
                    S[i] = RC[P[S[i]]]
                else
```

```
                    S[i] = LC[P[S[i]]]
                end if
            end if
            barrier

            for k = 1 to p do
                if k != i and P[S[k]] < S[i] then
                    S[i] = P[S[k]]
                end if
            end for
            barrier
        end for

        A[0] = S[0]
        for j = 1 to n-1 do
            A[j] = P[S[i]]
            P[S[i]] = -1
            barrier
        end for
end for
```

数组P来存储每个节点的父节点下标，每次处理完后将其置为-1，数组S来存储当前处理器正在处理的节点的下标。

所用模型：PRAM-CRCW. 需要共享存储，以及并发读和并发写。
时间复杂度:$O(logn)$

HW3

1、在PRAM-CREW模型上，用n个处理器在O(1)时间内求出数组A[1...n]={0,...,0,1,...,1}中第一个1的下标，写出并行伪代码
判断某个数是否左侧为0右侧为1即可（因为是CR，可以并发读）
伪代码如下：

```
algorithm Find_First_One(A[1...n], P[1...n])
输入：A[1...n] = {0,...,0,1,...,1}, n processors
输出：The index of the first 1 in A
Begin
    for each processor i do
        if A[i] == 1 then
            if A[i-1] == 0 && A[i+1] == 1 then
                return i
            end if
        end if
    end for
end
```

Ex7.3
(1)试分析算法7.3的复杂度

算法7.3为PRAM上对数划分算法
其中使用了二分查找，其时间复杂度为$O(logn)$

(2)令A=（0，1，2，7，9，11，16，17，18，19，23，24，25，27，28，30，33，34），B=（3，4，5，6，8，10，12，13，14，15，20，21，22，26，29，31）。试按照算法7.3，将其进行对数划分，并最终将他们归并

$m = 16$
$k(m) = m/logn = 16/4 = 4$
$rank(6 : A) = 3$
$rank(13 : A) = 6$
$rank(21 : A) = 10$
因此有四组成对划分:
$A_0 = (0, 1, 2), B_0 = (3, 4, 5, 6)$
$A_1 = (7, 9, 11), B_1 = (8, 10, 12, 13)$
$A_2 = (16, 17, 18, 19), B_2 = (14, 15, 20, 21)$
$A_3 = (23, 24, 25, 27, 28, 30, 33, 34), B_3 = (22, 26, 29, 31)$
成对归并:
$C_0 = (0, 1, 2, 3, 4, 5, 6)$
$C_1 = (7, 8, 9, 10, 11, 12, 13)$
$C_2 = (14, 15, 16, 17, 18, 19, 20, 21)$
$C_3 = (22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34)$
最终归并:
$C = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34)$

Ex7.6
(1)试分析算法7.9的总运算量W(n)
假设赋值也算入运算操作
Step1: n次
Step2: $1 + 2 + \ldots + n/2 = n - 1$次
Step3: $1 + 2 + \ldots + n/2 + n = 2n - 1$次
所以$W(n) = n + n - 1 + 2n - 1 = 4n - 2$

(2)假定序列为(1,2,3,4,5,6,7,8)，试用算法7.9求其前缀和
用SIMD_TC上非递归求前缀和算法

Step1:
初始化 B(0, j) = A(j):
B(0,1) = 1
B(0,2) = 2
B(0,3) = 3
B(0,4) = 4
B(0,5) = 5
B(0,6) = 6
B(0,7) = 7
B(0,8) = 8

Step2:
正向遍历，计算局部前缀和：
h = 1:
B(1,1) = B(0,1) + B(0,2) = 1 + 2 = 3
B(1,2) = B(0,3) + B(0,4) = 3 + 4 = 7
B(1,3) = B(0,5) + B(0,6) = 5 + 6 = 11
B(1,4) = B(0,7) + B(0,8) = 7 + 8 = 15
h = 2:
B(2,1) = B(1,1) + B(1,2) = 3 + 7 = 10
B(2,2) = B(1,3) + B(1,4) = 11 + 15 = 26

Step3:
反向遍历，计算全局前缀和：
h = 2:
C(2,1) = B(2,1) = 10

C(2,2) = B(2,2) = 26
h = 1:
C(1,1) = B(1,1) = 3
C(1,2) = C(2,1) = 10
C(1,3) = B(1,3) = 11
C(1,4) = C(2,2) = 26
h = 0:
C(0,1) = B(0,1) = 1
C(0,2) = C(1,1) = 3
C(0,3) = B(0,3) = 6
C(0,4) = C(1,2) = 10
C(0,5) = B(0,5) = 15
C(0,6) = C(1,3) = 21
C(0,7) = B(0,7) = 28
C(0,8) = C(1,4) = 36

因此序列 (1, 2, 3, 4, 5, 6, 7, 8) 的前缀和为 (1, 3, 6, 10, 15, 21, 28, 36)