清华大学软件学院

算法分析与设计基础

2020 年春季学期

作业 **11**

吴佳龙 班级：软件 83 学号：2018013418 June 20, 2020

11.1. (CLRS Exercises 27.2-3)

只需要将朴素的矩阵乘法中的第三重循环的冒险消除即可，采用分治。伪代码如下：

```
P−DOT(x,y,l,r):
    if l==r:
        return x[l]*y[l]
    mid = (l+r)/2
    a = spawn P−DOT(x,y,l,mid)
    b = P−DOT(x,y,mid+1,r)
    sync
    return a+b

P−SQUARE−MATRIX−MULTIPLY(A,B):
    n = A.rows
    let C be a new n times n matrix
    parallel for i = 1 to n
        parallel for j = 1 to n
            x = ith row of A
            y = jth column of B
            c[i][j] = P−DOT(x,y,1,n)
    return C
```

复杂度分析 对 i 的并行循环，递归树的深度是 $\Theta(\lg n)$；对 j 的并行循环，递归树的深度也是 $\Theta(\lg n)$；利用分治并行计算的点积，递归树的深度也是 $\Theta(\lg n)$；最终递归树的叶子节点的复杂度为 $\Theta(1)$，所以总的持续时间为

$$T_\infty = \Theta(\lg n) + \Theta(\lg n) + \Theta(\lg n) + \Theta(1) = \Theta(\lg n)$$

P-DOT 的串行化的复杂度仍然是 $T_1(n) = 2T_1(n/2) + \Theta(1) = \Theta(n)$，仅增加了常数因子。因此总的工作量仍是 $\Theta(n^3)$。

11.2. (CLRS Problem 27-2)

(a) 伪代码如下，并假设在调用前，C 已初始化为全 0：

```
P−MATRIX−MULTIPLY−RECUSIVE'(C,A,B):
    n = A.rows
    if n == 1
        c[1][1]+=a[1][1]*b[1][1]
    else
        partition A,B,C into n/2 times n/2 submatrices

        spawn P−MATRIX−MULTIPLY−RECUSIVE'(C11,A11,B11)
        spawn P−MATRIX−MULTIPLY−RECUSIVE'(C12,A11,B12)
        spawn P−MATRIX−MULTIPLY−RECUSIVE'(C21,A21,B11)
        P−MATRIX−MULTIPLY−RECUSIVE'(C22,A21,B12)
        sync

        spawn P−MATRIX−MULTIPLY−RECUSIVE'(C11,A12,B21)
        spawn P−MATRIX−MULTIPLY−RECUSIVE'(C12,A12,B22)
        spawn P−MATRIX−MULTIPLY−RECUSIVE'(C21,A22,B21)
        P−MATRIX−MULTIPLY−RECUSIVE'(C22,A22,B22)
        sync
```

(b) 该算法的工作量为

$$T_1(n) = 8T_1(n/2) + \Theta(1) = \Theta(n^3)$$

持续时间为

$$T_\infty(n) = 2T_\infty(n/2) + \Theta(1) = \Theta(n)$$

(c) 并行度为 $T_1/T_\infty = \Theta(n^2)$。当 $n = 1000$ 时，并行度为 10^6，而 P-MATRIX-MULTIPLY-RECURSIVE 的并行度为 10^7，是它的 10 倍。

11.3. 在 OpenMP 平台上实现并比较多线程的归并排序和快速排序算法。

见源代码和实验报告。