并行计算 HW3 及 HW4

PB18111697 王章瀚

2021 年 5 月 18 日

1.

在 PRAM-CREW 模型上, 用 n 个处理器在 O(1) 时间内求出数组 $A[1..n] = \{0, \cdots, 0, 1, \cdots, 1\}$ 中最先为 1 值的下标, 写出并行伪代码.

Algorithm 1 First-One-Index(A)

Require: 数组 $A[1..n] = \{0, \cdots, 0, 1, \cdots, 1\}$

```
1: function First-One-Index(A)
2:     for i = 1..n − 1 par- do
3:         if A[i] == 0 and A[i + 1] == 1 then
4:             return i + 1
```

7.3.

(1).

试分析算法 7.3 的时间复杂度.

先按步说明:

(1). 显然 O(1)

(2). 循环并行化后, 每个循环体内需要求 rank, 这最多需要 $\Theta(n)$, 但是如果能够给出 n 个进程, 就能在 O(1) 内判断出 $b_{i \log m}$ 在 A 中的 rank.

(3). 每个并行体内均是数据拷贝, 共需要 $\Theta(\log m + j(i+1) - j(i))$, 这其实是 $O(\log m + n)$ 的.(这是因为如果 a_1 大于 B 中所有元素, 则有一个进程要拷贝 n 个数据, 就算并行, 也需要等它完成它的任务, 这需要 $\Theta(n)$). 但如果 A 数据是均匀分布的, 就只需要 $\Theta(\log m + \frac{n}{m} \log m)$ 的时间.

因此总共需要 $O(\log m + n)$. 若 A 元素比较均匀, 则是 $\Theta(\log m + \frac{n}{m} \log m)$.

(2).

令 $A = \{0, 1, 2, 7, 9, 11, 16, 17, 18, 19, 23, 24, 25, 27, 28, 30, 33, 34\}, B = \{3, 4, 5, 6, 8, 10, 12, 13, 14, 15, 20, 21, 22, 26, 29, 31\}$. 试按算法 6.3 将其对数划分, 并最终将它们归并.

对此情况, $n = 18$, $m = 16$, $k(m) = m/\log m = 4$,

(1). $j(0) = 0$; $j(k(m)) = j(4) = 18$

(2). 并行得到

- $b_{1\log m} = b_4 = 6, j(1) = rank(b_4 : A) = 3$
- $b_{2\log m} = b_8 = 13, j(2) = rank(b_8 : A) = 6$
- $b_{3\log m} = b_{12} = 21, j(3) = rank(b_{12} : A) = 10$

(3). 并行得到分组结果:

- $A_0 = (0, 1, 2), B_0 = (3, 4, 5, 6)$
- $A_1 = (7, 9, 11), B_1 = (8, 10, 12, 13)$
- $A_2 = (16, 17, 18, 19), B_2 = (14, 15, 20, 21)$
- $A_3 = (23, 24, 25, 27, 28, 30, 33, 34), B_3 = (22, 26, 29, 31)$

分段归并即得:

- $(0, 1, 2, 3, 4, 5, 6)$
- $(7, 8, 9, 10, 11, 12, 13)$
- $(14, 15, 16, 17, 18, 19, 20, 21)$
- $(22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34)$

拼接起来即:

$$(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34)$$

7.6.

(1).

试分析算法 7.9 的总运算量 W(n)

逐步说明：

(1). 初始化 n 个进程计算量均为 $\Theta(1)$, 故总共是 $\Theta(n)$

(2). 对 h 从 1 到 $\log n$ 共 $\log n$ 次循环. 其内并行 $n/2^h$ 个进程, 每个经过 $\Theta(1)$ 的计算, 因此总共

$$\Theta(\sum_{h=1}^{\log n} n/2^h) = \Theta(n(1 - (\frac{1}{2})^{\log n})) = \Theta(n(1 - \frac{1}{n}) = \Theta(n)$$

(3). 类似地, h 共 $\Theta(\log n)$ 层循环, 其内 $n/2^k$ 个进程内部各需要 $\Theta(1)$ 的操作, 总共是

$$\Theta(\sum_{h=0}^{\log n} n/2^h) = \Theta(2n(1 - (\frac{1}{2})^{\log n})) = \Theta(2n)$$

因此总运算量是 $W(n) = \Theta(n + n + 2n) = \Theta(4n)$

(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

按步说明:

(1). 初始化步, $B(0,1)=1, B(0,2)=2, \cdots, B(0,8)=8$, 即 B 如下表:

1	2	3	4	5	6	7	8

(2). 正向遍历时, 总将 $B(h,j)=B(h-1,2j-1)*B(h-1,2j)$

1	2	3	4	5	6	7	8
3		7		11		15	
10				26			
36							

(3). 反向遍历时, 即能根据 B 求出 C:

1	3	6	10	15	21	28	36
3		10		21		36	
10				36			
36							

C 的第一行即为前缀和

2.6

2.6 一个 $N=2^n$ 个节点的洗牌交换网络如图 2.36 所示。试问:此网节点度、网络直径和网络对剖宽度分别是多少?

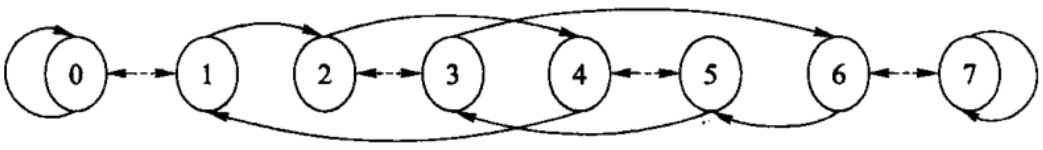

图 2.36 $N=8$ 的洗牌交换网络

- 节点度: 入射边和出射边之和. 每个节点在交换网络有两个度, 在洗牌网络有两个度, 总共节点度是 4.
- 网络直径: 网络中任何两个节点之间的最长距离. 如 N=8, 则应为节点路径 $0 \to 1 \to 2 \to 3 \to 6 \to 7$ 是最长的之一, 其长度为 5. 对于 $N=2^n$ 也类似, 最远的应该是节点 0 到节点 N-1, 路径应为交换, 洗牌不断轮流(即刚开始加 1, 然后每次乘 2 加 1), 因此其长度为 $2n-1$.

- 网络对剖宽度: 对分网络各半所需最少边数. 如分为 $\{0,1,2,3\}$ 和 $\{4,5,6,7\}$, 中间要划分开需要去掉最中间的 4 条边($2\to4, 3\to6, 4\to1, 5\to3$). 而对于 $N=2^n$ 的情况, 只要考虑 0 到 $2^{n-1}-1$ 和 2^{n-1} 到 2^n-1 有多少条边即可. 因此对剖宽度为 2^{n-1}.

2.7

2.7 一个 $N=(k+1)2^k$ 个节点的蝶形网络如图 2.37 所示。试问:此网节点度、网络直径和网络对剖宽度分别是多少?

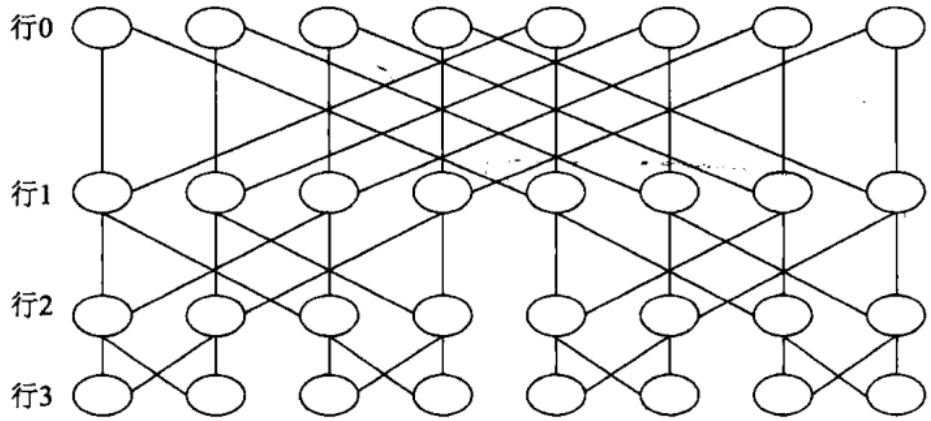

图 2.37 $k=3$ 的蝶形网络

- 节点度: 入射边和出射边之和. 本网络是无向图. 第0行和最后一行每个节点都是 2 个边, 其节点度是 2; 而中间的行节点度都是 4.
- 网络直径: 网络中任何两个节点之间的最长距离. 从行0第一个节点到最后一个节点需要先传播到行k再回到行0, 是最长路径的之一, 长度为 $2k$. (图中情况 $2k=6$)
- 网络对剖宽度: 对分网络各半所需最少边数. 从中间对剖下去, 需要去掉 $2\times2^{k-1}=2^k$ 条边, 因此对剖宽度是 2^k. (图中情况 $2^k=8$)

2.15

2.15 一到多个人通信又称之为单点散播(Single-Node Scatter),它与一到多播送不同之处是,此时源处理器有 p 个信包,每一个去向一个目的地(见图 2.32(c))。图 2.41 示出了 8 个处理器的超立方上单点散射的过程。试证明:使用 SF 和 CT 方式在超立方上施行一到多个人通信的通信时间为

$$t_{\text{one-to-all-pers}}=t_s\log p+mt_w(p-1) \qquad (2.14)$$

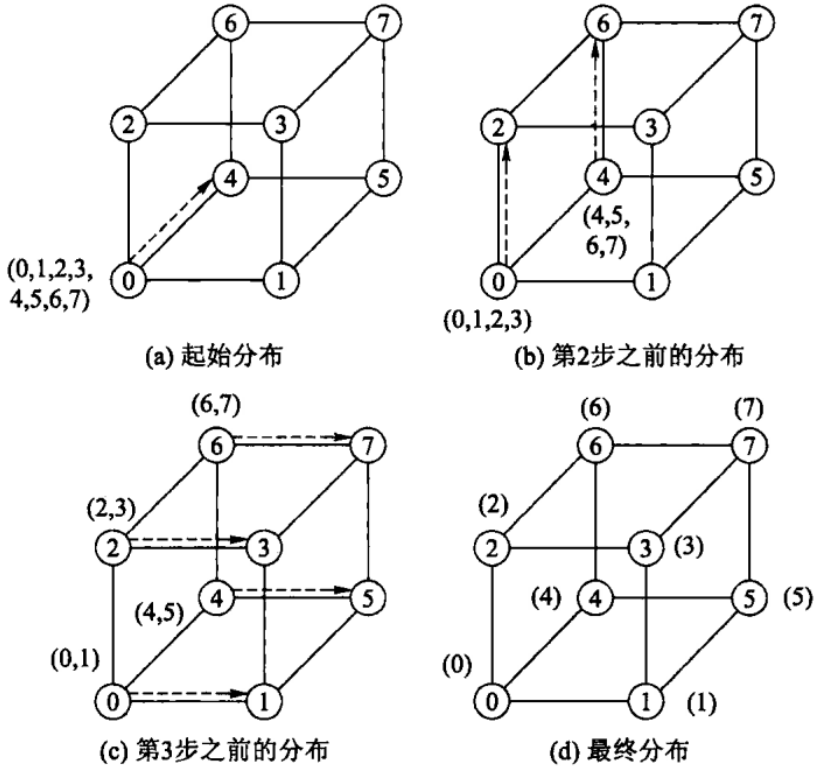

图 2.41　8 个处理器的超立方上单点散射过程

依据公式有:

$$t_{comm}(SF) = t_s + (mt_w + t_h)l$$

$$t_{comm}(CT) = t_s + mt_w + lt_h$$

其中 t_s 是启动时间, t_h 是节点延迟时间, t_w 是字传输时间, 信包大小为 m, 链路数为 l.

具体到此题中, 可以忽略 t_h. 而由于本题中的传播过程每步都是传播一跳, 因此使用 CT 并不会有优势, 每次都仍然需要建立链路.

另外, 每步传输都将可以 drop 一半的包, 每次传播也都涉及了 2 倍的节点. 因此第 i 步, 需要传播包大小为 $\frac{p}{2^i}$, 总共传播 $\log p$ 次(即 $i = 1, 2, \cdots, \log p - 1$, 因为是超立方, p 为 2 的幂). 因此第 i 次传播用的时间为 $t_s + mt_w\frac{p}{2^i}$

由此可知, 总时间为

$$\begin{aligned} t_{one-to-all-pers} &= t_s \log p + \sum_{i=1}^{\log p} mt_w \frac{p}{2^i} \\ &= t_s \log p + \sum_{i=1}^{\log p} mt_w 2^i \\ &= t_s \log p + mt_w 2^{\log p} - 1 \\ &= t_s \log p + mt_w(p-1) \end{aligned}$$