并行计算作业 4 & 5

傅申 PB20000051

2.6.

节点度 4, 包括交换的 2 个度和洗牌的两个度;

网络直径 对图中 $N = 8$ 的洗牌交换网络, 节点 0 和 7 之间的路径是最长的, 距离为 5, 最短路径为 $0 \to 1 \to 2 \to 3 \to 6 \to 7$, 因此网络直径为 5; 对 $N = 2^n$ 的洗牌交换网络, 仍然是节点 0 和 $2^n - 1$ 之间的路径最长, 最短路径为交换/洗牌轮流进行的路径, 即 $0 \to 1 \to ... \to \left(2^i - 2\right) \to \left(2^i - 1\right) \to ... \to (2^n - 2) \to (2^n - 1)$, 长度 (网络直径) 为 $2n - 1$.

网络对剖宽度 对图中 $N = 8$ 的洗牌交换网络, 将节点分为 $\{0, 1, 2, 4\}$ 和 $\{3, 5, 6, 7\}$ 两部分, 需要移去的边只有 2 条 ($2 \leftrightarrow 3$, $4 \leftrightarrow 5$) 为最小值, 因此对剖宽度为 2; 对 $N = 2^n$ 的情况, 对剖宽度有上界 $O(\frac{N}{n})$, 如下表, 证明见 MIT 的 Theory of Parallel Systems (SMA 5509) 课程的 Lecture 18 的 Note 最后一章.

N	**对剖宽度**	**去掉的边**
2	1	$0 \leftrightarrow 1$
4	1	$1 \leftrightarrow 2$
8	2	$2 \leftrightarrow 3, 4 \leftrightarrow 5$
16	4	$2 \leftrightarrow 3, 8 \leftrightarrow 9, 10 \leftrightarrow 11, 14 \leftrightarrow 15$

2.7.

节点度 行 0 和行 k 的节点度为 2, 中间行的节点度为 4.

网络直径 直径为 $2k$, 对应的路径为行 0 的第一个节点到最后一个节点的路径, 需要先到行 k 再回到行 0.

网络对剖宽度 从中间对剖下去, 需要去掉 $2 \times 2^{k-1} = 2^k$ 条边, 即为对剖宽度.

2.15.

如图中过程所示, 每步传输都是传播一条, 每次传递的信包量都会减半, 而涉及的节点数量都会加倍, 因此只需要 $\lg p$ 步传输即可完成单点散播. 因此, SF 方式的传输时间为

$$
\begin{aligned}
t_{\text{one-to-all-pers}}(\text{SF}) &= \sum_{i=1}^{\lg p} \left(t_s + \left(\frac{mp}{2^i} t_w + t_h \right) l \right) \\
&= t_s \lg p + \sum_{i=1}^{\lg p} \frac{mp}{2^i} t_w \qquad (\text{忽略 } t_h \text{ 并且 } l = 1) \\
&= t_s \lg p + m t_w (p - 1)
\end{aligned}
$$

同理, 对于 CT 方式, 传输时间为

$$
\begin{aligned}
t_{\text{one-to-all-pers}}(\mathrm{CT}) &= \sum_{i=1}^{\lg p}\left(t_s + \frac{mp}{2^i}t_w + lt_h\right) \\
&= t_s \lg p + \sum_{i=1}^{\lg p}\frac{mp}{2^i}t_w \qquad (\text{忽略 } t_h \text{ 并且 } l=1) \\
&= t_s \lg p + mt_w(p-1)
\end{aligned}
$$

因此, 两种方式的传输时间相同, 均为 $t_{\text{one-to-all-pers}} = t_s \lg p + mt_w(p-1)$.

9.S1.

如下

```
begin
  (1) for all P[i, j] par-do
      (1.1) C[i, j] = 0
      (1.2) for k = 0 to √p - 1 do
            (1.2.1) C[i, j] += A[i, (i + j + k) % √p] * B[(i + j + k) % √p, j]
            end for
      end for
end
```

其中 (1.1) 的时间为 $O(1)$, 而 (1.2) 执行了 $\sqrt{p}$ 次 $(n / \sqrt{p}) \times (n / \sqrt{p})$ 子块乘法, 时间为 $\sqrt{p} \times (n / \sqrt{p})^3 = n^3 / p$, 因此时间复杂度为 $O(n^3 / p)$.

9.9.

(2.1) 的运行时间为 t_a, (2.2) 迭代了 n 轮, 其中每一轮需要读存储器 3 次, 进行一次乘法和一次加法, 再写入存储器一次, 每一轮时间为 $4t_a + 2t_c$, 因此总的并行运行时间为 $(4n+1)t_a + 2nt_c$.