

考前复习

魏立斐

第二章 系统互连与基本通信操作

使用SF进行多到多播送

- 环

- 步骤

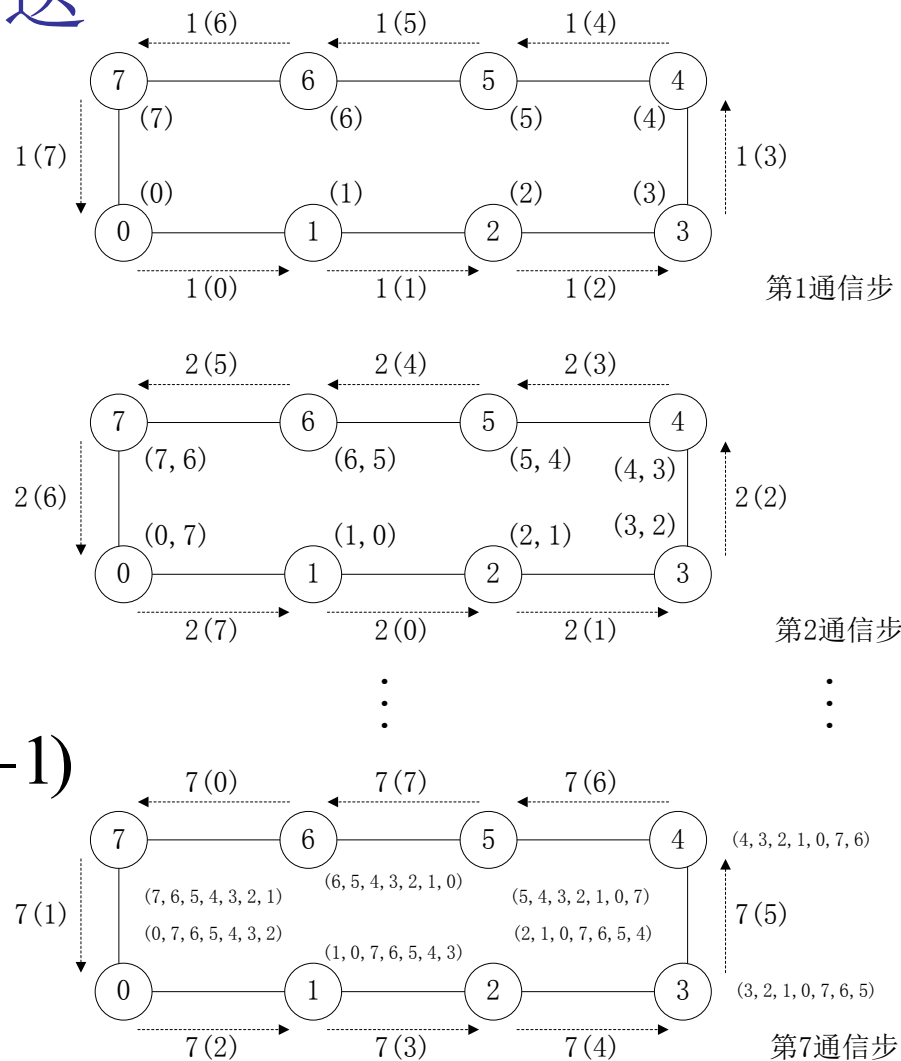
- 同时向右(或左)播送

- 刚接收到的信包

- 示例

- 通讯时间:

$$t_{all-to-all}(SF) = (t_s + mt_w)(p-1)$$



第二章 系统互连与基本通信操作

使用SF进行多到多播送

- 环绕网孔

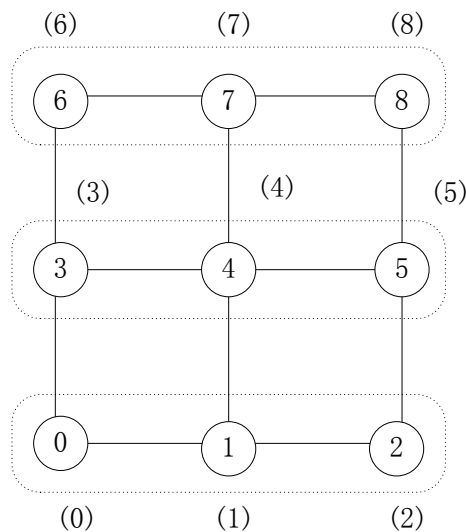
- 步骤

- 先进行行的播送

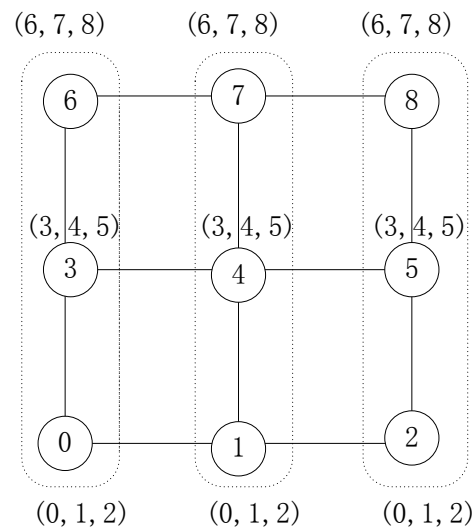
- 再进行列的播送

- 注意：信包变大

- 示例



(a)



(b)

- 通讯时间:

$$\begin{aligned} t_{all-to-all}(SF) &= (t_s + mt_w)(\sqrt{p} - 1) + (t_s + m\sqrt{p} \cdot t_w)(\sqrt{p} - 1) \\ &= 2t_s(\sqrt{p} - 1) + mt_w(p - 1) \end{aligned}$$

第二章 系统互连与基本通信操作

使用SF进行多到多播送

- 超立方

- 步骤

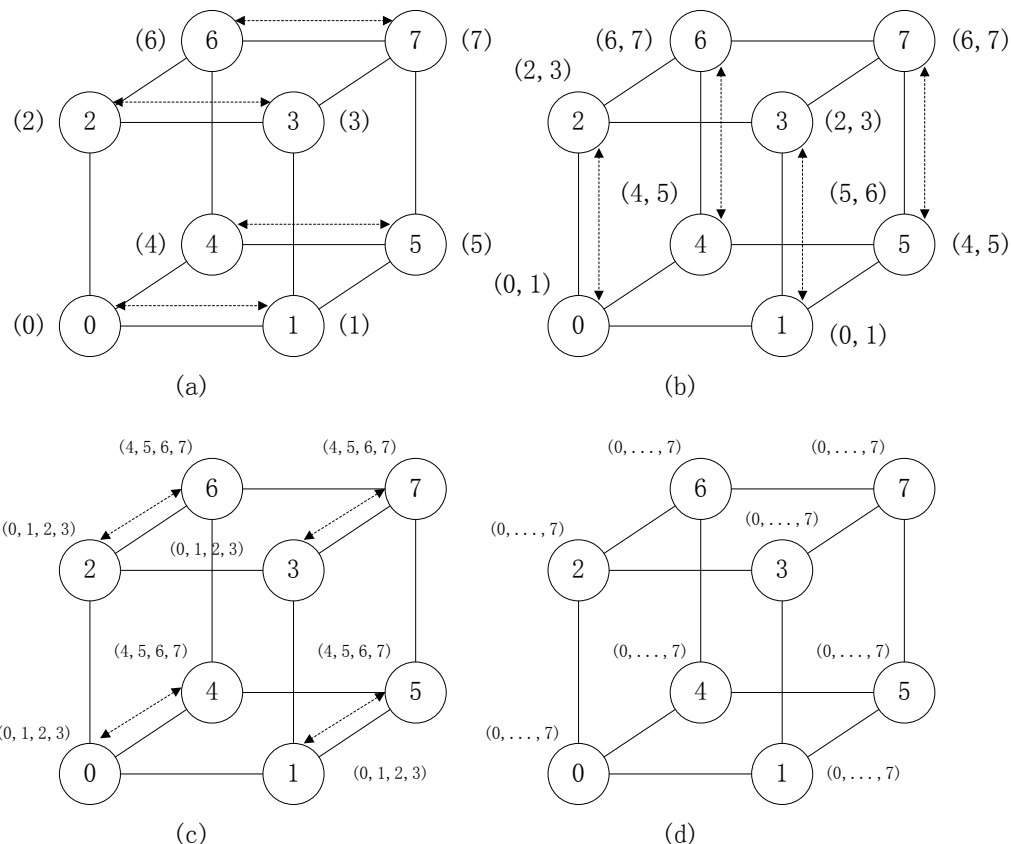
- 依次按维进行多到多的播送

- 示例

- 通讯时间:

$$t_{all-to-all}(SF) = \sum_{i=1}^{\log p} (t_s + 2^{i-1} m t_w)$$

$$= t_s \log p + m t_w (p - 1)$$



Amdahl 定律:

- 出发点:
 - 科学计算中实时性要求很高，时间是个关键的因素，而固定不变的计算负载；
 - => 在一定的计算负载下，为达到实时性可利用增加处理器加快执行速度
 - 固定的计算负载分布在多个处理器上的，
 - => 增加了处理器就加快执行速度，从而达到了加速的目的。
- 1967年，Amdahl提出固定负载的加速公式:

$$S = \frac{W_s + W_p}{W_s + W_p / p}$$

Amdahl定律

- 固定负载的加速公式： $S = \frac{W_s + W_p}{W_s + W_p / p}$
- $W_s + W_p$ 可相应地表示为 **$f + (1-f)$**

$$S = \frac{f + (1-f)}{f + \frac{1-f}{p}} = \frac{p}{1 + f(p-1)}$$

当 $p \rightarrow \infty$ 时，上式极限为： **$S = 1/f$**

例：一个串行程序**94%**的执行时间花费在一个可并行化的函数中。现使其并行化，问该并行程序在**10**个处理机上执行所能达到的加速比是多少？

第六章 并行算法的基本设计策略

■ 快速排序算法的串行实现

输入：无序序列(A_q, \dots, A_r) 输出：有序序列(A_q, \dots, A_r)

Proc. QUICKSORT (A, q, r)

 Begin

 if $q < r$ then

 (1) $x = A_q$ // 赋为主元

 (2) $s = q$

 (3) for $i = q + 1$ to r do

 if $A_i \leq x$ then

 (i) $s = s + 1$

 (ii) swap(A_s, A_i)

 endif

 (4) swap(A_q, A_s)

 (5) QUICKSORT(A, q, s)

 (6) QUICKSORT($A, s + 1, r$)

 endif

End

算法实例： $A = \{4, 1, 3, 2, 5\}$,

快排序算法的并行化

- 算法 PRAM-CRCW上的快排序二叉树构造算法

输入：序列 (A_1, \dots, A_n) 和 n 个处理器

输出：供排序用的一棵二叉排序树

Begin

```
(1)for each processor i do (2)repeat for each processor i<>root do
  (1.1) root=i              if  $(A_i < A_{f_i}) \vee (A_i = A_{f_i} \wedge i < f_i)$  then
  (1.2)  $f_i = \text{root}$         (2.1) $LC_{f_i} = i$ 
  (1.3)  $LC_i = RC_i = n + 1$   (2.2)if  $i = LC_{f_i}$  then exit else  $f_i = LC_{f_i}$  endif
  end for                  else
                          (2.3) $RC_{f_i} = i$ 
                          (2.4)if  $i = RC_{f_i}$  then exit else  $f_i = RC_{f_i}$  endif
                          endif
                          end repeat
End
```


第六章 并行算法的基本设计策略

程序说明

- 序列 (A_1, \dots, A_n) 和 n 个处理器
- 处理器 P_i 保存：
 - 元素 A_i
 - LC_i 和 RC_i 分别记录给定主元的左儿子和右儿子
 - F_i 是存有主元的处理器号
- A_{root} 是第一个主元
- 那些元素小于 A_{f_i} 的处理器将其号码写入 LC_i ，大于 A_{f_i} 的处理器将号码写入 RC_i
- 并发操作只写了两个值，它们成为了下一次迭代所需主元的处理器号。

算法实例：A={4, 1, 3, 2, 5}，设编号小的处理器写优先

P_1 A_1 4	P_2 A_2 1	P_3 A_3 3	P_4 A_4 2	P_5 A_5 5
f_1 LC_1 RC_1 1 6 6	f_2 LC_2 RC_2 1 6 6	f_3 LC_3 RC_3 1 6 6	f_4 LC_4 RC_4 1 6 6	f_5 LC_5 RC_5 1 6 6
	$A_2 < Af_2 = A_1$ $LCf_2 = LC_1 = 2$ $i = 2 = LCf_2 = 2$ exit	$A_3 < Af_3 = A_1$ $LCf_3 = LC_1 = 3$ $i = 3 \neq LCf_2 = 2$ $f_3 = LCf_3 = LC_1 = 2$	$A_4 < Af_4 = A_1$ $LCf_4 = LC_1 = 4$ $i = 4 \neq LCf_2 = 2$ $f_4 = LCf_4 = LC_1 = 2$	$A_5 > Af_5 = A_1$ $RCf_2 = RC_1 = 5$ $i = 5 = RCf_5 = 5$ exit
f_1 LC_1 RC_1 1 2 5	f_2 LC_2 RC_2 1 6 6	f_3 LC_3 RC_3 2 6 6	f_4 LC_4 RC_4 2 6 6	f_5 LC_5 RC_5 1 6 6
		$A_3 > Af_3 = A_2$ $RCf_3 = RC_2 = 3$ $i = 3 = RCf_3 = 3$ exit	$A_4 > Af_4 = A_2$ $RCf_4 = RC_2 = 4$ $i = 4 \neq RCf_3 = 3$ $f_4 = RCf_4 = RC_2 = 3$	
f_1 LC_1 RC_1 1 2 5	f_2 LC_2 RC_2 1 6 3	f_3 LC_3 RC_3 2 6 6	f_4 LC_4 RC_4 3 6 6	f_5 LC_5 RC_5 1 6 6

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/795340113031011333>