并 行 计 算

——结构•算法•编程

主讲教师：谢磊

第三篇 并行数值算法

第八章 基本通讯操作

第九章 稠密矩阵运算

第十章 线性方程组的求解

第十一章 快速傅里叶变换

第八章 并行数值算法

8.0 预备知识

8.1 选路方法与开关技术

8.2 单一信包一到一传输

8.3 一到多播送

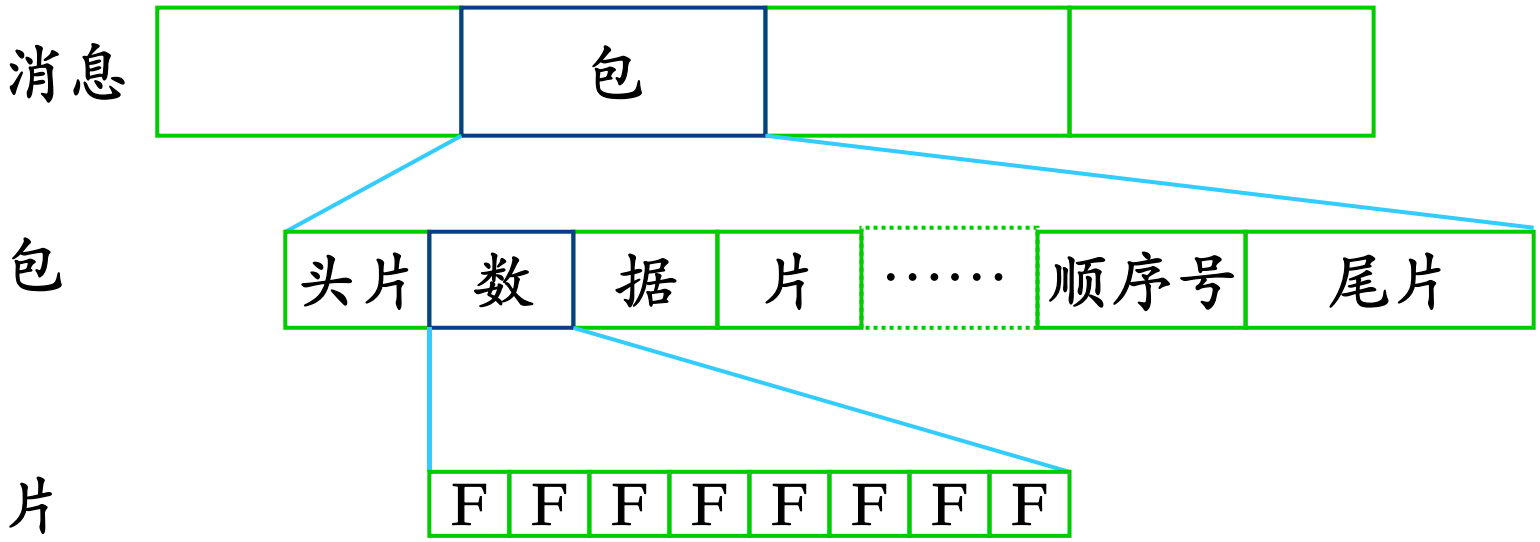
8.4 多到多播送

预备知识

* 选路(Routing)
  * 又称为选径或路由。产生消息从发源地到目的地所取的路径，要求具有较低通讯延迟、无死锁和容错能力。应用于网络或并行机上的信息交换。
* 消息、信包、片
  * 消息(Message)：是在多计算机系统的处理接点之间传递包含数据和同步消息的信息包。它是一种逻辑单位，可由任意数量的包构成。
  * 包(Packet)：包的长度随协议不同而不同，它是信息传送的最小单位，64-512位。
  * 片(Flit)：片的长度固定，一般为8位。

预备知识

* 消息、信包、片的相互关系



预备知识

* 一些术语
  * 信道带宽b：每个信道有w位宽和信号传输率$f = 1/t$ (t是时钟周期)，$b = wf$ *bits/sec*
  * 节点和开关的度：与节点和开关相连的信道数目
  * 路径：信包在网络中走过的开关和链路(link)序列
  * 路由长度或距离：路由路径中包括的链路(link)数目
* 信包传输性能参数
  * 启动时间t_s(startup time)：准备包头信息等
  * 节点延迟时间t_h(per-hop time)：包头穿越相邻节点的时间
  * 字传输时间t_w(transfer time)：传输每个字的时间
  * 链路数l、信包大小m

预备知识

* 选路算法的三种机制
  * 基于算术的: 开关中具有简单的算术运算功能，如维序选路;
  * 基于源地址的: 在源点时就将沿路径的各个开关的输出端口地址$p_0,p_1,\cdots,p_n$包在信包的头部，每个开关只是对信包头的输出端口地址进行剥离;
  * 基于查表的: 开关中含有一个选路表，对信包头中的选路域查出输出端口地址。

预备知识

* 选路方式

- 信包
  - 存储－转发（*Store and Forward*）
  - 虫孔（*Wormhole*）
- 线路：用交换机

- 静态：在选路开始时所有的信息都已到达网络
- 动态：信包可在任意时刻到达网络

- 联机(*online*)：没有事先计算好的路径
- 脱机(*offline*)：事先算好传输路径

- 一到一（单播）
- 一到一（置换）：每个处理器开始时最多发送一条信包，每条信包有且仅有一个目的地；
- 多到一（集中）
- 一到多（多播）
- 一到所有（广播、组播）
- 多到多（会议）

第八章 并行数值算法

8.0 预备知识

8.1 选路方法与开关技术

8.2 单一信包一到一传输

8.3 一到多播送

8.4 多到多播送

8.1 选路方法与开关技术

8.1.1 选路方法

8.1.2 开关技术

选路方法

* 分类
  * 最短路径/非最短路径(贪心选路/随机选路),
  如维序选路是贪心的，二阶段维序选路是随机的
  * 确定选路/自适应选路(寻径确定/寻径视网络状况)
* 维序选路(Dimension-Ordered Routing):
  一种确定的最短路径选路
  * 二维网孔中的维序选路: X-Y选路
  * 超立方中的维序选路: E-立方选路

选路方法

* X-Y选路算法
  * 算法8.1：二维网孔上的X-Y选路算法

begin

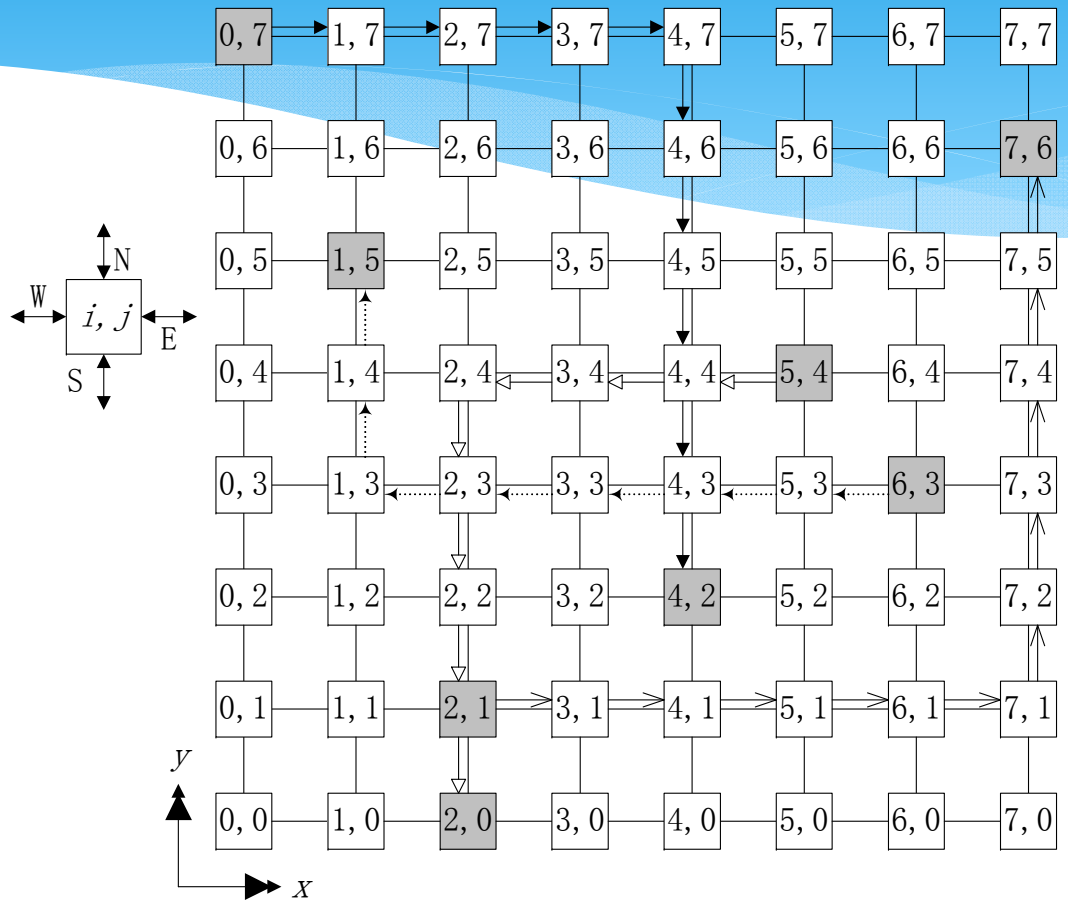
step1: 沿X方向将信包送至目的地处理器所在的列

step2: 沿Y方向将信包送至目的地处理器所在的行

end

选路方法

* 例8.1



4(源;目的)对: (2, 1;7, 6) ⟶ (5, 4;2, 0) ⟶

(0, 7;4, 2) ⟶ (6, 3;1, 5) ⟶

图8. 1

选路方法

* E-立方选路算法
  * 路由计算：$s_{n-1}s_{n-2}\cdots s_1s_0$(源地址)

    异或 $\oplus$ $d_{n-1}d_{n-2}\cdots d_1d_0$(目的地址)

    $r_{n-1}\,r_{n-2}\cdots r_1\,r_0$(路由值)
  * 路由过程：

$s_{n-1}s_{n-2}\cdots s_1s_0 \rightarrow s_{n-1}s_{n-2}\cdots s_1s_0 \oplus r_0 \rightarrow$

$s_{n-1}s_{n-2}\cdots s_1s_0 \oplus r_1 \rightarrow \cdots$

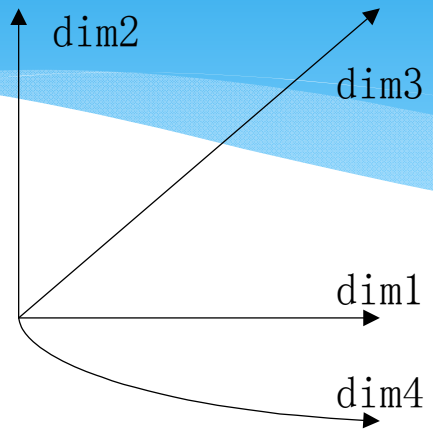
选路方法

* 例8.2

0110(S)

1101(D)

1011(R)



源： S=0110
目的： D=1101
路径：
0110 → 0111 → 0101 → 1101

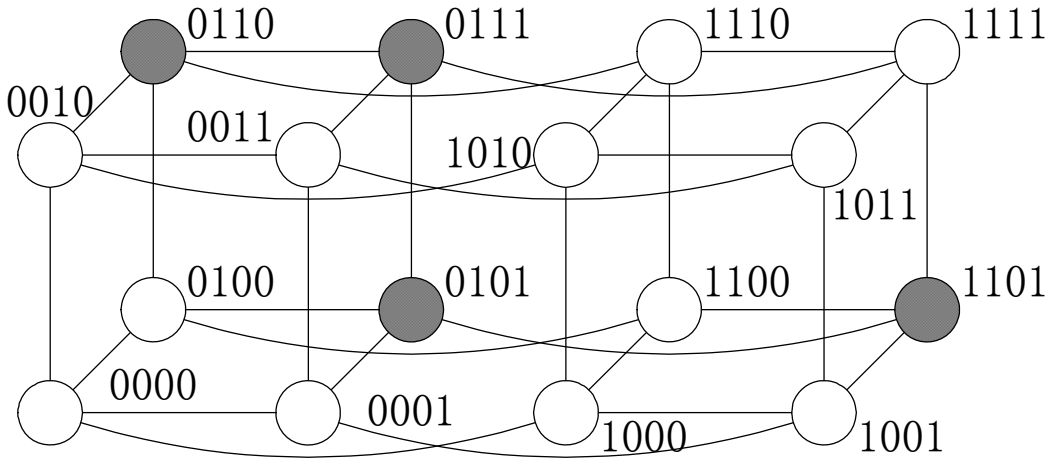


图8.2

选路方法

* 算法8.2：超立方网络上的E-立方选路算法
* 输入：待选路的信包在源处理器中
* 输出：将源处理器中的信包送至其目的地

Begin

(1) for i=1 to n do

$r_i = s_{i-1} \oplus d_{i-1}$

endfor

(2) i=1,V=S

(3) while $i \leq n$ do

(3.1) if $r_i = 1$ then 从当前节点V选路到节点为$V \oplus 2^{i-1}$ endif

(3.2) i=i+1

endwhile

End

8.1 选路方法与开关技术

8.1.1 选路方法

8.1.2 开关技术

开关技术

* 存储转发(Store-and-Forward)选路
  * 消息被分成基本的传输单位----信包(Packet), 每个信包都含有寻径信息;
  * 当一个信包到达中间节点A时，A把整个信包放入其通信缓冲器中，然后在选路算法的控制下选择下一个相邻节点B，当从A到B的通道空闲并且B的通信缓冲器可用时，把信包从A发向B;
  * 信包的传输时间: $t_{comm}(\mathrm{SF}) = t_s + (mt_w + t_h)l = O(ml)$

缺点:

* 每个结点必须对整个消息和信包进行缓冲，缓冲器较大;
* 网络时延与发送消息所经历的节点数成正比

开关技术

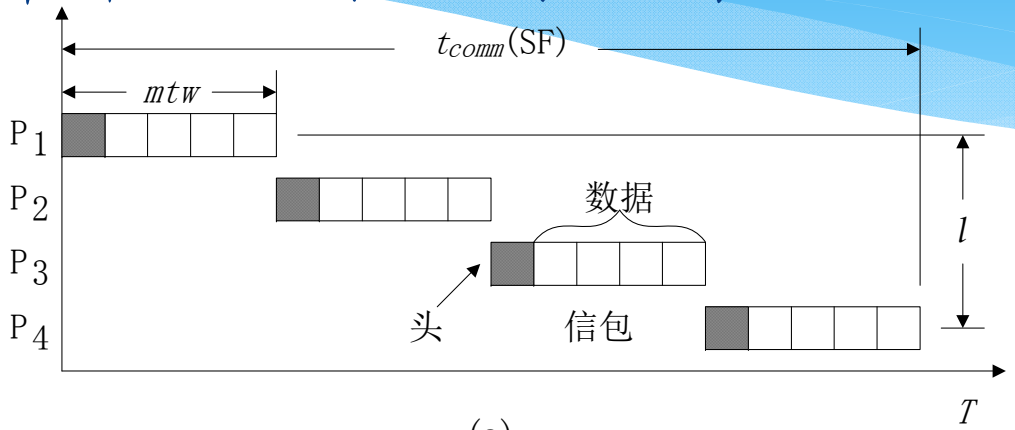
* 切通(Cut Through)选路
  * 在传递一个消息之前，就为它建立一条从源结点到目的结点的物理通道。在传递的全部过程中，线路的每一段都被占用，当消息的尾部经过网络后，整条物理链路才被废弃。
  * 传输时间: $t_{comm}(\mathrm{CT}) = t_s + mt_w + lt_h = O(m+l)$

缺点:

* 物理通道非共享
* 传输过程中物理通道一直被占用

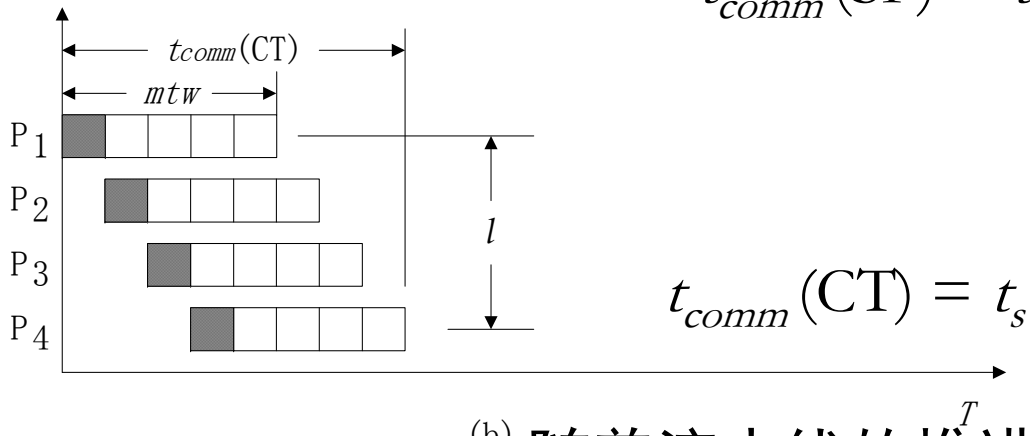
开关技术

* 几种开关技术的时空图



(a)

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



(b)

$$t_{comm}(\mathrm{CT}) = t_s + (m+l-1)t_w + lt_h$$

随着流水线的推进，(l-1)tw可以忽略。

图8.4

开关技术

* 通信时间 t_{comm}(SF) 与 t_{comm}(CT)的近似
  * 一般情况下，ts是不容忽略的，同时tw通常也比th大得多，因此，上述 t_{comm}(SF) 与 t_{comm}(CT)可近似写为：

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

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

开关技术

* 虫孔(Wormhole)选路
  * Dally于1986年提出，利用了前二种方法的优点，减少了缓冲区，提高了物理通道的利用。
  * 首先把一个消息分成许多很小的片，消息的头片包含了这个消息的所有寻径信息。尾片是一个其最后包含了消息结束符的片。中间的片均为数据片；
  * 片是最小信息单位。每个结点上只需要缓冲一个片就能满足要求；
  * 用一个头片直接牵引一条从输入链路到输出链路的路径的方法来进行操作。每个消息中的片以流水的方式在网络中向前“蠕动”。每个片相当于Worm的一个节，“蠕动”以节为单位顺序地向前爬行。当消息的尾片向前“蠕动”一步后，它刚才所占用的结点就被放弃了。

开关技术

* 虫孔(Wormhole)选路

优点：

(1)每个结点的缓冲器的需求量小，易于用VLSI实现；

(2)较低的网络传输延迟。存储转发传输延迟基本上正比于消息在网络中传输的距离；Wormhole与线路开关的网络传输延迟正比于消息包的长度，传输距离对它的影响很小（消息包较长时的情况）；

(3)通道共享性好、利用率高；

(4)易于实现Multicast和Broadcast。

第八章 并行数值算法

8.0 预备知识

8.1 选路方法与开关技术

8.2 单一信包一到一传输

8.3 一到多播送

8.4 多到多播送

单一信包一到一传输

* 距离*l*的计算：对于*p*个处理器
  * 一维环形： $l \le \lfloor p/2 \rfloor$
  * 带环绕Mesh($\sqrt{p} \times \sqrt{p}$)： $l \le 2\lfloor \sqrt{p}/2 \rfloor$
  * 超立方： $l \le \log p$
* t_{comm}(SF)的计算(可由(8.1b)式得到)
  * 一维环形： $t_{comm}(SF) = t_s + t_w \cdot m \cdot \lfloor p/2 \rfloor$
  * 带环绕Mesh： $t_{comm}(SF) = t_s + 2t_w \cdot m \cdot \lfloor \sqrt{p}/2 \rfloor$
  * 超立方： $t_{comm}(SF) = t_s + t_w \cdot m \cdot \log p$
* t_{comm}(CT)的计算(可由(8.2b)式得到)

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

第八章 并行数值算法

8.0 预备知识

8.1 选路方法与开关技术

8.2 单一信包一到一传输

8.3 一到多播送

8.4 多到多播送

8.3 一到多播送

8.3.1 SF模式

8.3.2 CT模式

一到多播送—SF模式

* 环
  * 步骤：①先左右邻近传送;②再左右二个方向同时播送
  * 示例：

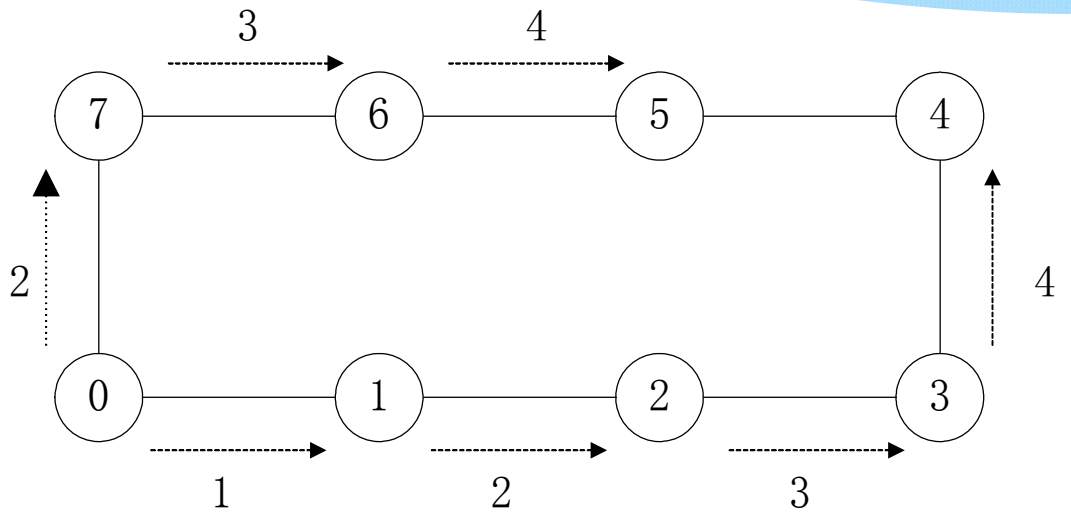


图8.5

  * 通讯时间：$t_{one-to-all}(SF) = (t_s + mt_w)\lceil p/2 \rceil$

一到多播送—SF模式

* 环绕网孔
  * 步骤：①先完成一行中的播送;②再同时进行各列的播送
  * 示例：
    共4步(2步行、2步列)

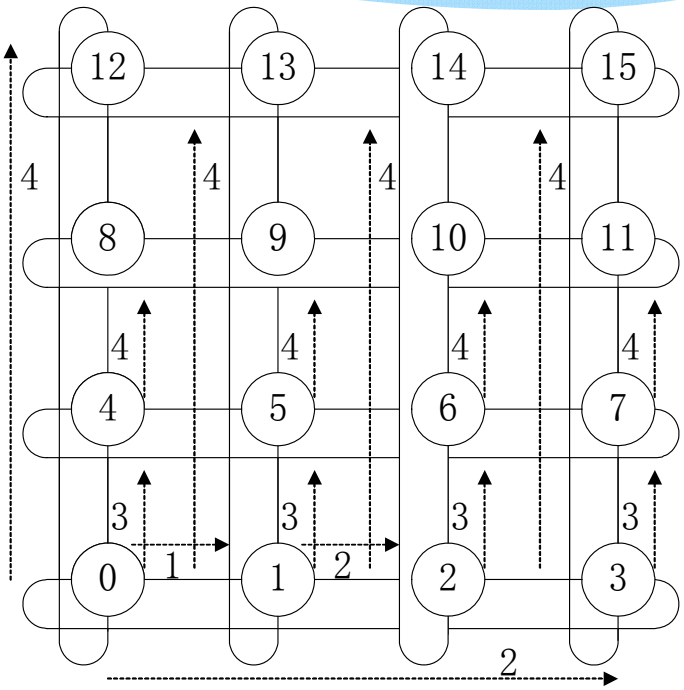
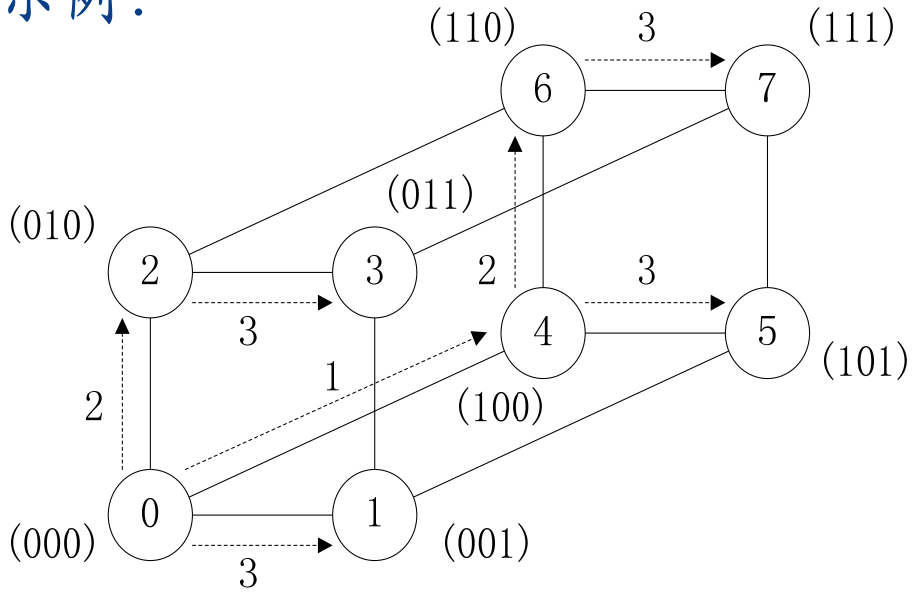


图 8.6

* 通讯时间：$t_{one-to-all}(SF) = 2(t_s + mt_w)\left\lceil \frac{\sqrt{p}}{2} \right\rceil$

一到多播送—SF模式

* 超立方
  * 步骤：从低维到高维，依次进行播送;
  * 示例：



  * 通讯时间：$t_{one-to-all}(SF) = (t_s + mt_w)\log p$

8.3 一到多播送

8.3.1 SF模式

8.3.2 CT模式

一到多播送—CT模式

* 环
  * 步骤：

(1)先发送至p/2远的处理器;

(2)再同时发送至$p/2^2$远的处理器;

……

(i)再同时发送至$p/2^i$远的处理器;

  * 示例： 图8.8

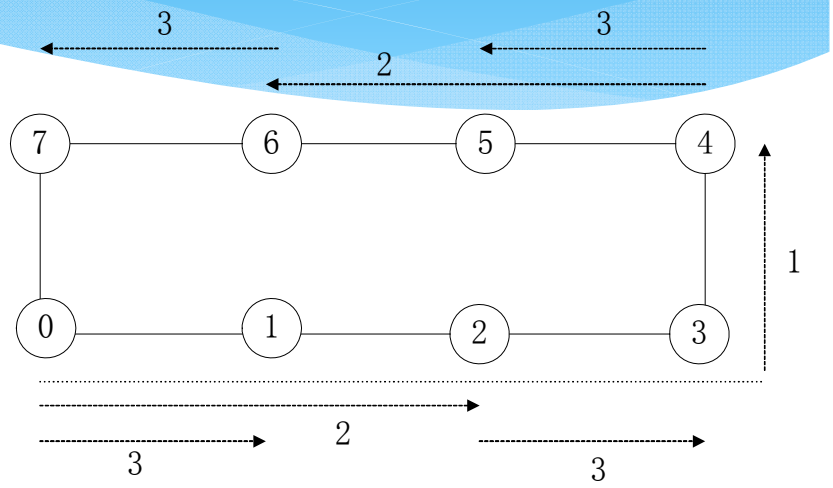


图 8.8

  * 通讯时间：

$$t_{one-to-all}(CT) = \sum_{i=1}^{(8.2)\log p}(t_s + mt_w + t_h p/2^i)$$

$$= t_s \log p + mt_w \log p + t_h(p-1)$$

$$\approx (t_s + mt_w)\log p \qquad (t_h\text{可忽略时})$$

一到多播送—CT模式

* 网孔
  * 步骤：

(1)先进行行播送;

$$// \; t_s \log\sqrt{p} + mt_w \log\sqrt{p} + t_h(\sqrt{p}-1)$$

(2)再同时进行列播送;

$$// \; t_s \log\sqrt{p} + mt_w \log\sqrt{p} + t_h(\sqrt{p}-1)$$

  * 示例： 图8.9

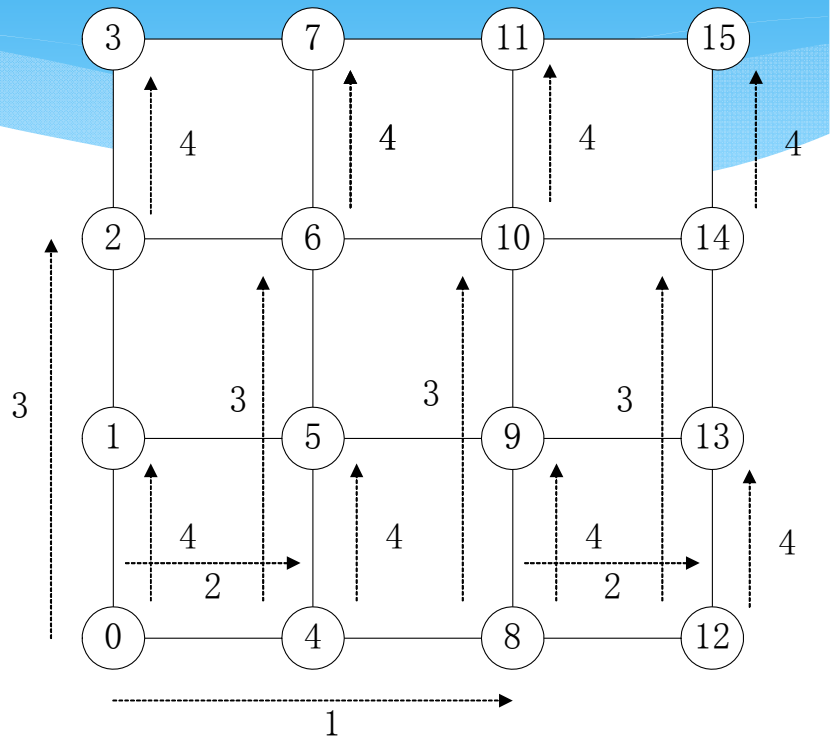


图 8.9

  * 通讯时间：

$$t_{one-to-all}(CT) = 2(t_s \log\sqrt{p} + mt_w \log\sqrt{p} + t_h(\sqrt{p}-1))$$

$$= (t_s + mt_w)\log p + 2t_h(\sqrt{p}-1)$$

一到多播送—CT模式

* 超立方
  * 步骤：

    依次从低维到高维播送，d-立方，d=0,1,2,3,4…；

  * 通讯时间：$t_{one-to-all}(CT) = (t_s + mt_w)\log p$

第八章 并行数值算法

8.0 预备知识

8.1 选路方法与开关技术

8.2 单一信包一到一传输

8.3 一到多播送

8.4 多到多播送

8.3 一到多播送

8.3.1 SF模式

8.3.2 CT模式

多到多播送—SF模式

* 环
  * 步骤：

同时向右(或左)播送刚接收到的信包

  * 示例： 图8.10

  * 通讯时间：

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

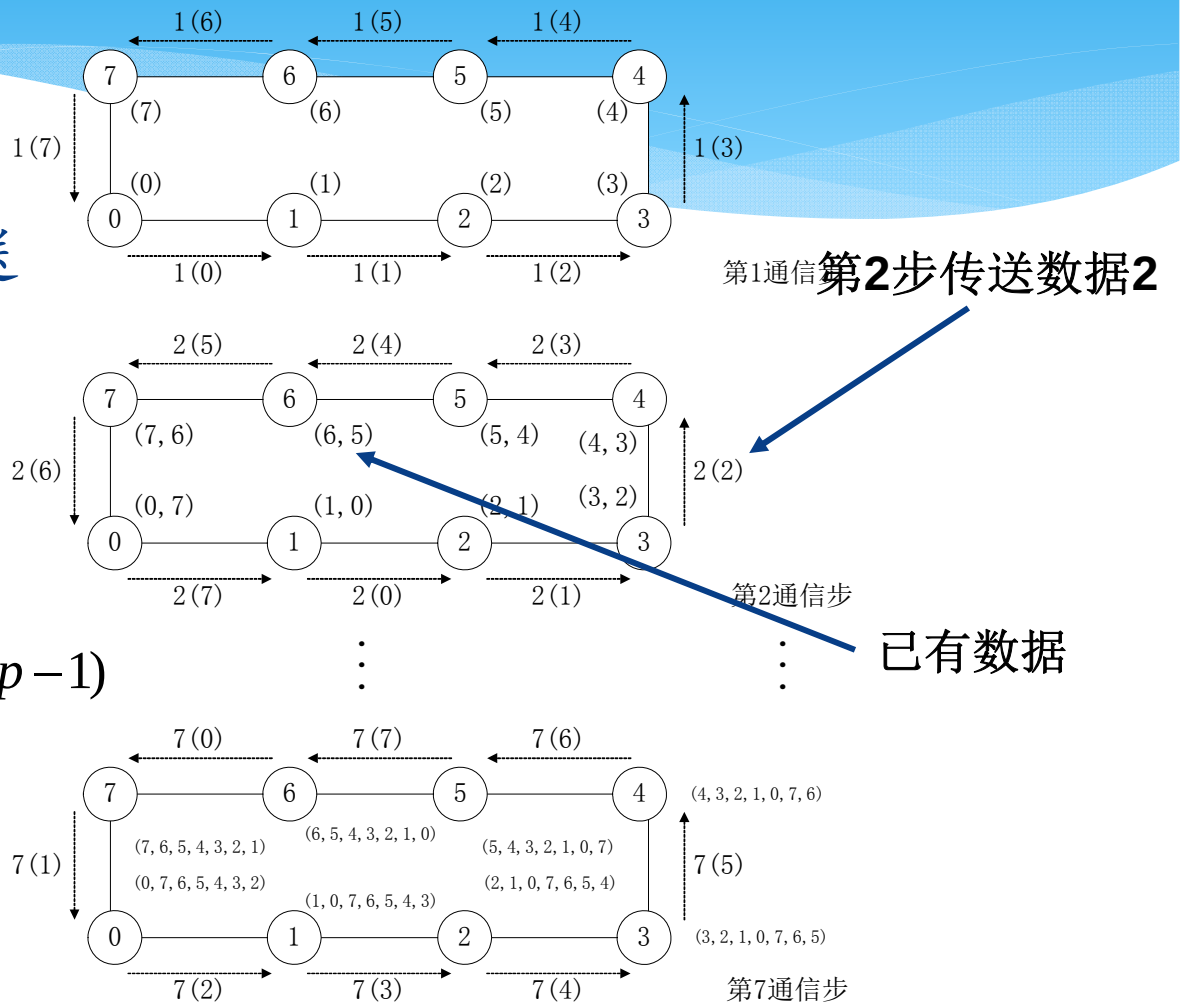


图8. 10

多到多播送—SF模式

* 环绕网孔
  * 步骤：

(1)先进行行的播送；

(2)再进行列的播送；

  * 示例： 图8.11

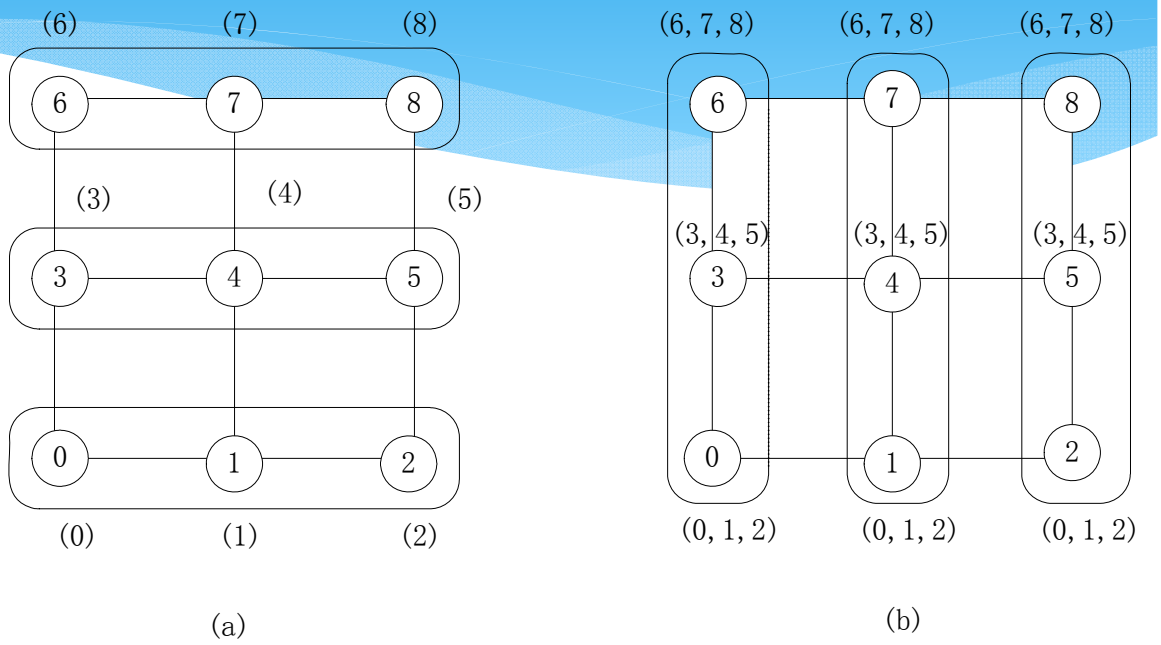


图8. 11

  * 通讯时间：

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

$$= 2t_s(\sqrt{p} - 1) + mt_w(p - 1)$$

多到多播送—SF模式

* 超立方
  * 步骤：
    依次按维进行
    多到多的播送；
  * 示例： 图8.12
  * 通讯时间：

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

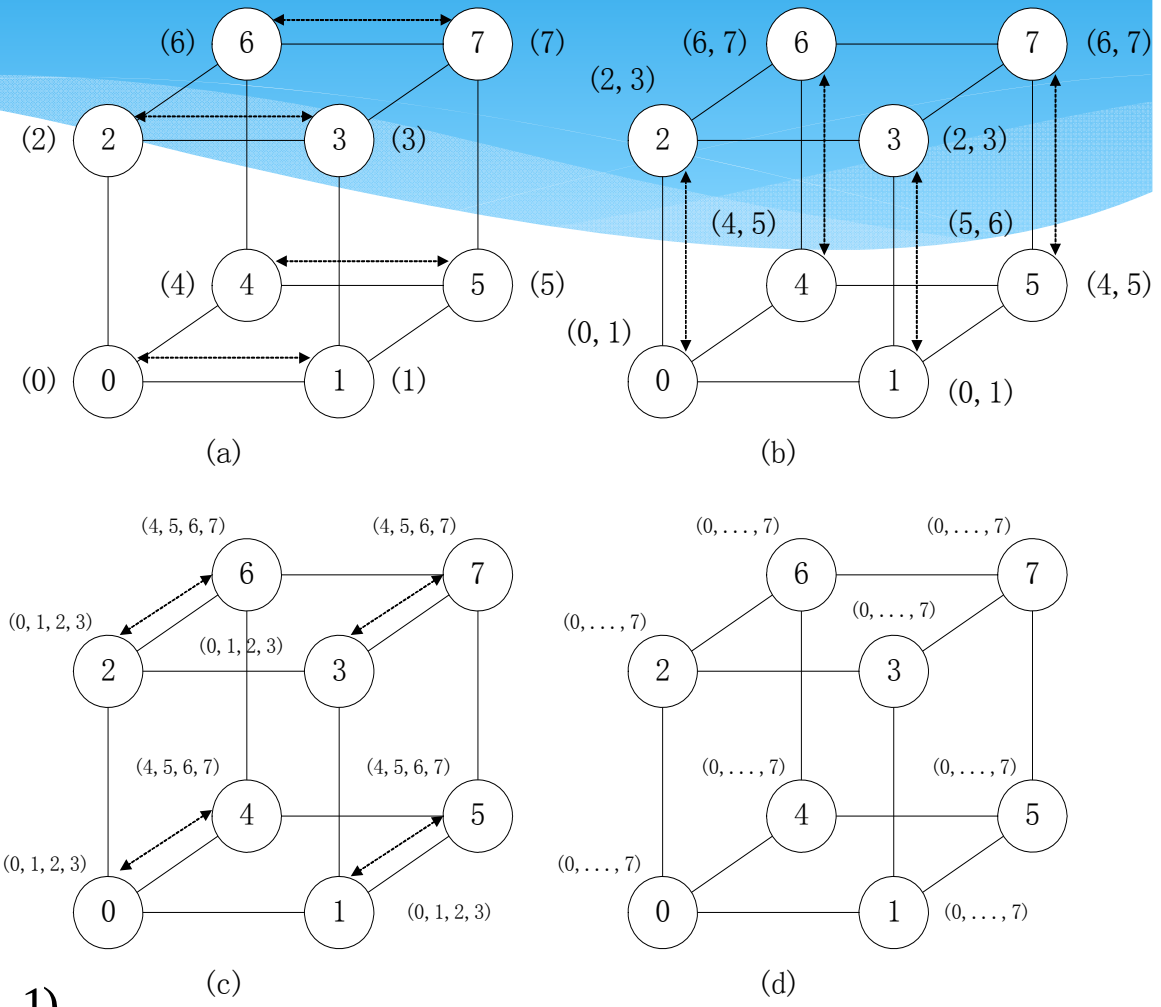


图8. 12

8.3 一到多播送

8.3.1 SF模式

8.3.2 CT模式

多到多播送—CT模式

* $t_{all-to-all}(CT) = t_{all-to-all}(SF)$
* 使用一到多的策略会造成链路竞争

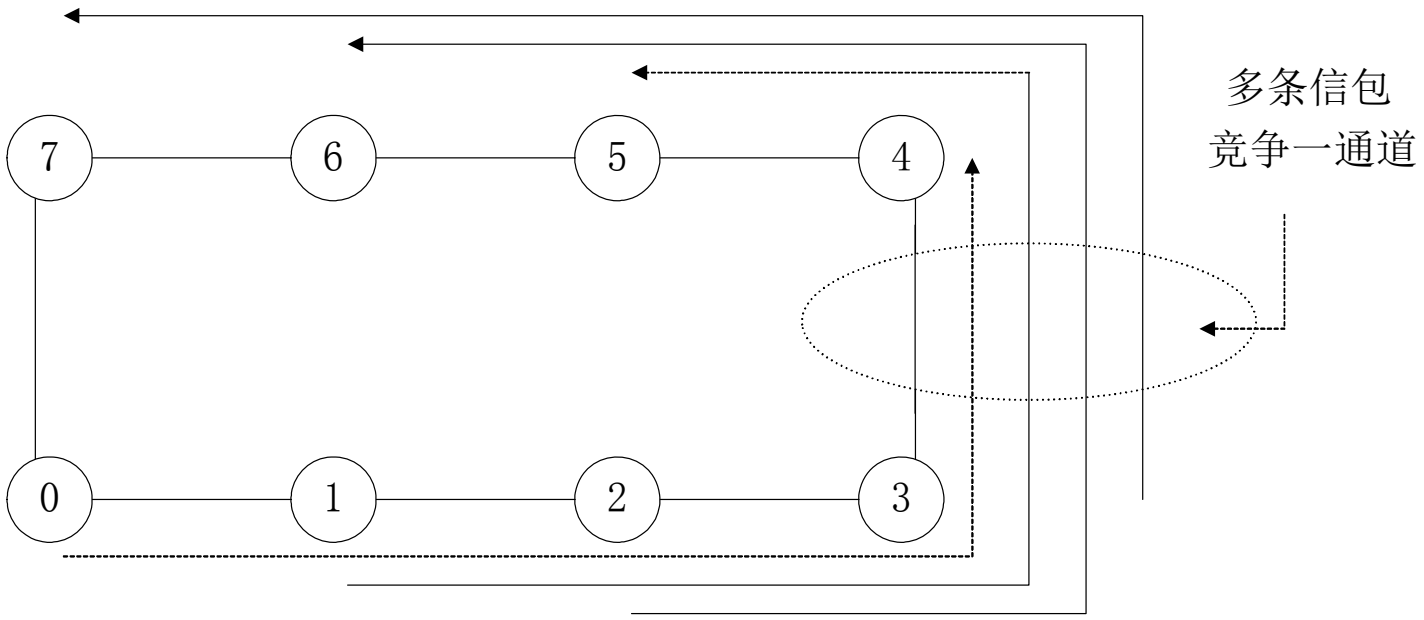


图8. 13