[文章编号] 1007-7405(2005)04-0329-04

基于输入交换队列的双向图匹配调度算法的协议模型表示

鄂大伟

(集美大学计算机工程学院, 福建 厦门 361021)

[摘要] 针对基于定长分组 (信元) 双向图匹配的输入队列交换机调度算法, 提出了算法的统一符号表示与描述方法及其实例.

[关键词] 交换; 双向图匹配; 调度算法

[中图分类号] TP393.01 [文献标识码] A

0 引言

目前, 基于 Crossbar输入队列交换结构的高速交换机和路由器的研究日益活跃. 输入队列技术的优点是其交换结构简单, 内部交换速度要求较低, 交换机只需以连接速度进行操作, 易于实现. 然而输入队列的调度策略较为复杂, 因为调度策略不仅要解决输出信元竞争同一输出端口时所产生的冲突问题, 还要解决共享同一输入端口所带来的冲突问题. 传统的输入队列技术一般使用 FIFO队列, 由于存在着链头 HOL (head of line) 阻塞现象, 基于输入队列定长分组交换机在单播和均匀业务流量条件下其吞吐率只能达到 58.6%[1]. 输入队列多播交换机受到 HOL阻塞的影响更为严重, 因为队首信元可能希望输出到多个输出端口, 从而使其吞吐率更低. 目前, 对支持单播通信的输入队列交换机的研究已经取得进展. 研究表明, 只要采用适当的输入队列技术 (如虚输出队列 VOQ)[2], 交换机的吞吐量可达到 100%, 从而使基于输入队列交换机的研究进入新的阶段.

在基于双向匹配图的调度算法中, 实际上是采用了一个协议过程来匹配. 在提出的各种调度算法中, 在表述其输入/输出端的匹配过程时, 有其各自的描述方法, 大多数算法都没有以符号化 (或结构化) 的形式给出, 给算法的性能分析和程序实现带来了不便. 笔者在分析各种算法的基础上, 提出了双向图匹配算法状态信息的符号表示模型, 使算法的匹配迭代过程清晰明了, 易于描述, 为算法的程序实现及性能模拟分析提供了新的手段.

1 基于双向匹配图的调度算法

分组 (或信元) 调度算法又称为仲裁算法, 它是通过解决在每一个时隙中发生在相同输出端口的冲突问题, 来控制队列对交换结构的访问, 抽象出来就是一个双向图 (bipartite graph) 匹配问题.

为了建立端口间的连接, 调度算法的主要任务是解决如图 1所示的双向图匹配问题. 在图 1中, V中的元素称为顶点, 任一顶点满足 $1 \leqslant i \leqslant M$和 $1 \leqslant j \leqslant N$, M和 N为顶点数, 顶点对间的连线称

[收稿日期] 2005-08-29
[基金项目] 福建省教育厅科技资助项目 (JA04234)
[作者简介] 鄂大伟 (1956-), 男, 教授, 从事计算机网络、软件工程建模等方向的研究.

为边（edge）. 这时无向图 G 可定义为：$G = [V, E]$，其中 V 是非空顶点的集合，E 是连接顶点的边的集合. 连接非空顶点的边有一个标识为 ω_i 的值，称为该边的权（weight）. 匹配 M 是 E 的任一子集，在 M 中，没有两条边具有相同的顶点.

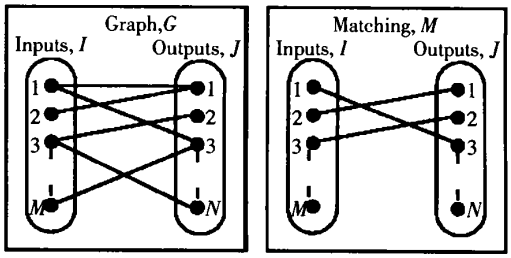

图 1 双向图 G 和匹配 M

Fig.1 Bipartite graph G and matching M

基于定长分组交换机的匹配算法（matching algorithms）就是研究输入与输出端口之间建立双向匹配的问题，即在尽可能少的时间内，在输入端至输出端之间实现最大匹配，并建立端口间的连接以进行信元交换. 文献 [3 4] 提出了许多用于最大双向图匹配的算法，给出了一些重要的和有意义的结论. 但这些算法的易实现性和时间复杂度是明显不同的，而这一点对于高速连接的交换引擎是非常重要的.

在基于双向匹配图的调度算法中，都采用了一个协议的过程来匹配. 它是一种在输入端和输出端采用握手协议进行连接的方法，这个过程可以用一个协议图来表示，如图 2所示. 一般由以下 3个步骤组成：

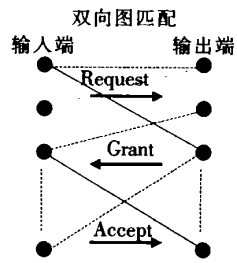

图 2 双向图匹配的 3个步骤

Fig.2 The three steps of bipartite graph matching

1）Request（请求）：输入端以广播方式向输出端发出请求信号（Request）；

2）Grant（响应）：由于一个输出端可能会收到来自多个输入端口的 Request信号，每一个输出端按照调度算法独立地从多个请求中选择一个，并向输入端返回 Grant信号；

3）Accept（确认）：由于输入端可能会收到来自多个输出端口的 Grant信号，每个输入端按照调度算法选择其中的一个 Accept信号予以确认.

通过以上 3个步骤，输入端与输出端便建立起了匹配连接.

现有的各种调度算法的功能不同，主要表现在第 2步和第 3步的方式不同，也就是输出端选择一个输入端予以响应，或输入端选择一个 Grant予以确认的方式不同.

2 双向图匹配算法的符号表示

对于具有 N 个输入端和 N 个输出端的 $N \times N$ 交换机，各输入端口依次用 In[0] … In[$N-1$] 表示，各输出端口依次用 Out[0] … Out[$N-1$] 表示.

第 i 个输入端口 In[i] 的状态信息用下列符号表示：

1）定义表 $R_i[0] \cdots R_i[N-1]$，当 In[i] 向输出端 Out[k] 发出 Request $R_i[k] = 1$，否则，$R_i[k] = 0$；

2）定义表 $Gd_i[0] \cdots Gd_i[N-1]$，当 In[i] 收到 Out[k] 的 Grant信号时，$Gd_i[k] = 1$，否则，$Gd_i[k] = 0$；

3）定义变量 A_i，如果 In[i] 接受 Out[k] 的响应，$A_i = k$，否则，$A_i = -1$.

类似地，第 k 个输出端口 Out[k] 也包含下面信息：

1）定义表 $Rd_k[0] \cdots Rd_k[N-1]$，当 Out[k] 收到 In[i] 请求时，$Rd_k[i] = 1$，否则，$R_i[k] = 0$；

2）定义变量 G_k，当 Out[k] 向 In[i] 发出响应信号，$G_k = i$，否则，$G_k = -1$；

3）定义变量 Ad_k，如果 Out[k] 的响应信号被接受，$Ad_k = 1$，否则，$Ad_k = 0$

输出端必须用随机的或旋转的方式检查请求输入端口，具体选择响应哪一个输入端口请求，为了说明

这一过程，笔者在每个输出端口用两个变量 g_k 和 g_k'，变量 g_k 表示在一次迭代中优先响应的输入端，g_k' 表示在随后的迭代中优先响应的输入端. 同样地，输入端有选择接受哪一个响应的过程，笔者在输入端口用两个变量 a_i 和 a_i' 分别表示在一次迭代中优先接受的输出端和在随后的迭代中优先接受的输出端.

3　算法描述

各种调度算法的不同在于输出端和输入端选择响应和接受的方式上，也就是在计算 g_k 和 a_i 的值时的计算方法不同. 按照笔者提出的统一算法的符号表示，本文完成了如上所述算法的统一形式描述，为算法的程序实现及性能模拟分析提供了新的手段. 限于篇幅，以下仅举两例说明.

3.1　iSLIP算法的表示

iSLIP算法[5]的 3个步骤如下：

1）请求：每个未匹配的输入端向所有可能的输出发送请求.

2）响应：如果未匹配输出端接收到请求，从最高优先级的端口开始，固定地从下一个出现的 Request请求信号中选择当前优先级最高的元素. 输出端通知所有输入端口它的请求是否被响应. 只有在第一次迭代时第三步响应信号才被输入端接受，在输出端指向最高优先级位置的指针 g_i 按模 N 增加，指向向输入端发出响应信号的下一个输入端位置.

3）接受：如果未匹配的输入端收到 Grant信号后，它从最高优先级的端口位置开始，固定地从下一个出现的 Grant信号中选择，只有在第一次迭代建立连接时，输入端指向最高优先级指针元素的指针 a_i 按模 N 增加，指向接收输出端口信号的下一个位置.

多次迭代 iSLIP算法在单次迭代 iSLIP算法的基础上增加一个迭代次数记数器 M，输入端口 In[i] 和输出端口 Out[k] 在每次迭代中执行下面操作：

$Rd_k[i] = R_i[k]$

$g_k = g_k'$

$a_i = a_i'$

$G_k = i$　if ($Rd_k[i] = 1$ and $g_k = i$) OR ($Rd_k[i] = 1$ and $Rd_k[j] = 0$　where $g_k \leqslant j < i \pmod N$)

$Gd_i[k] = 1$　if $G_k = i$

$A_i = k$　if ($Gd_i[k] = 1$ and $a_i = k$) OR ($Gd_i[k] = 1$ and $Gd_i[j] = 0$　Where $a_i \leqslant j < k \pmod N$)

If ($Ad_k = 1$) and ($M = 0$), $g_k' = G_k + 1 \pmod N$, $a_i' = A_i + 1 \pmod N$

Else $g_k' = g_k$, $a_i' = a_i$

3.2　多次迭代 FIRM 算法表示

多次迭代 FIRM （Fcfs In Round robin Matching）算法[6]修改多次迭代 iSLIP算法优先级指针的更替规则，消除了 iSLIP算法的不公平性. 多次迭代 FIRM 需要修改多次迭代 iSLIP算法：

1）请求：每个未匹配的输入端向所有可能的输出发送请求.

2）响应：如果未匹配输出端接收到请求，从最高优先级的端口开始，固定地从下一个出现的 Request请求信号中选择当前优先级最高的元素. 输出端通知所有输入端口它的请求是否被响应. 只有在第一次迭代时第三步准许信号才被输入端接受，在输出端指向最高优先级位置的指针 g_i 按模 N 增加 1，指向向输入端发出准许信号的下一个输入端位置. 如果在第一次迭代时第三步准许信号未被输入端接受，在输出端指向最高优先级位置的指针 g_i 指向准许的输入端位置.

3）接受：如果未匹配的输入端收到 Grant信号后，它从最高优先级的端口位置开始，固定地从下一个出现的 Grant信号中选择，只有在第一次迭代建立连接时，输入端指向最高优先级指针元素的指针 a_i 增加 1 (模 N)，指向接收输出端口信号的下一个位置. 输入端口 In[i] 和输出端口 Out[k] 在

每次迭代中执行下面操作：

$Rd_k[i] = R_i[k]$

$g_k = g_k'$

$a_i = a_i'$

$G_k = i$ if $(Rd_k[i] = 1$ and $g_k = i)$ OR $(Rd_k[i] = 1$ and $Rd_k[j] = 0$ where $g_k \leqslant j < i \pmod N)$

If $M = 0$, $g_k' = G_k$

$Gd_i[k] = 1$ if $G_k = i$

$A_i = k$ if $(Gd_i[k] = 1$ and $a_i = k)$ OR $(Gd_i[k] = 1$ and $Gd_i[j] = 0$ Where $a_i \leqslant j < k \pmod N)$

If $(Ad_k = 1)$ and $(M = 0)$, $g_k' = G_k + 1 \pmod N$, $a_i' = A_i + 1 \pmod N$

If $(Ad_k = -1)$ and $(M = 0)$, $g_k' = G_k \pmod N$

Else $g_k' = g_k$, $a_i' = a_i$

由算法表示可看出，FIRM算法比 iSLIP算法实现更公平的连接，且时延减小，不会出现端口饿死现象，一个请求最多 N^2 个时隙后就会得到服务.

[参考文献]

[1] Ge Nong, Jogesh K Muppala, Mounir Hamdi. Analysis of Nonblocking ATM Switches with Multiple Input Queues [J]. IEEE/ACM Transactions on Networking, 1999, 7(1): 60-63.

[2] Tamir Y, Frazier G. Dynamically-Allocated multi-queue buffer for VLSI communication switches [J]. IEEE Transaction on Computers, 1992, 41(6): 725-737.

[3] Nick McKeown, Thomas E Anderson. A Quantitative Comparison of Scheduling Algorithms for Input-Queued Switches [J]. Computer Networks and ISDN Systems, 1998, 30(24): 2 309-2 326.

[4] 鄂大伟. 基于多 FIFO输入队列交换结构的迭代匹配调度算法分析与比较 [J]，计算机工程与应用，2001, 11(37): 77-82.

[5] Nick McKeown. The iSLIP Scheduling Algorithm for Input-Queued Switches [J]. IEEE/ACM Transactions On Networking, 1999, 7(2): 188-201.

[6] Serpanos DN, Antoniadis PI. FIRM: A class of distributed scheduling algorithms for high-speed ATM switches with multiple input queues [J]. Proceedings of IEEE INFOCOM, 2000, (3): 548-555.

Presentation of Protocol Model for Bipartite Graph Match Scheduling Algorithm Based on Input Switching Queues

E Da-wei

(School of Computer Engineering, Jimei University, Xiamen 361021, China)

Abstract: Based on fixed length packet (cell) of bipartite graph match algorithm for input queue switches, the paper shows and describes the method and its instance after putting forward the unified symbol of the algorithm.

Key words: switching; bipartite graph match; scheduling algorithm

(责任编辑 朱雪莲)