计算机科学2000Vol. 27№. 5

一种应用于CCC的智能广播模型*)

A New Broadcasting Model for CCC

胡宁宁 陈道蓄

(南京大学计算机科学与技术系 计算机软件新技术国家重点实验室 南京210093)

Abstract The topology of CCC has been widely adapted in the tightly-coupled systems for its particular characteristics. In CCC, the performance of broadcasting is still an open issue. Usually, there are two models: classical and universal lists broadcasting models. But both of these models have some limitations in terms of complexity and performance. They either let each node store many transmission lists which use substantial unnecessary local memory, or use prescribed sequential selection which makes it lack adaptability and difficult to design. In this paper, the authors employ a new broadcasting model—ILBM—to design the broadcasting algorithm in CCC. This model proposes a selection algorithm that improves the adaptability and unifies the transmission lists, which ensure high performance broadcasting with small extra spatial cost, high adaptability, and convenience in implementation.

Keywords Broadcasting, Model, Algorithm, Cube-Connected Cycle(CCC)

1 引言

随着VLSI技术的发展,人们现在已经有能力将许多处理器集成在一块芯片上来实现一个高性能的多处理器系统[2~7]。对多处理器系统的拓扑结构人们已进行了大量研究[8,9],其中,超立方体结构是一种较具吸引力的拓扑结构,因为现实中的许多问题,比如离散傅立叶变换、排序等在多处理器并行系统中求解时都可以找到一些与超立方体有较好对应的算法。但在一个k维超立方体中,每个处理器结点都有k条链接,因此并不适合于VLSI技术的实现,为此人们曾提出过用混洗交换网(shuffle-exchange network)[10]结构替代之,然而Preparata与Vuilemin提出的CCC(Cube-Connected Cycles,立方体连接环)[11]则是一个更好的结构,这种结构具有以下特点:(1)每一处理器的链接数目不超过3;(2)链接数目的大幅度减小并没有引起处理时间的明显增大,因而处理性能并没有明显恶化;(3)CCC的整体结构适合于用VLSI技术实现。因此,CCC结构被广泛认为是一种很有前途的多处理器拓扑结构。

在多处理器并行计算系统中,全局广播是一种使用频繁且对许多并行算法相当关键的操作,广播性能的好坏直接影响了诸多并行算法的执行效率,对于紧耦合系统中的广播算法的设计,可分为硬件设计与软件设计两种,其中硬件设计就是指通过在硬件上改进系统中各结点的连接关系来改善广播的操作,而软件设计则是设计高效算法来达到提高效率的目的。本文主要讨论软件设计方面的问题。在软件设计的广播模型中,一般有两种:传统广播模型(CLBM)[1]及全局表模型(ULBM)[3],但它们都由于没有充分发挥软件的功能而具有较大的局限性。其中CLBM因存储多张表而不必要地耗费大量的系统资源;ULBM则在一张表上呆板地使用一种预先规定好的路由选择策略而使得该模型的适应性较差并且难于设计,针对于此,本文提出了一种新的广播模型—智能表广播模型(Intelligent List Broadcasting Model,ILBM),这种模型引入了结点算法而增强了网络广播操作的自适应性,统一了网络中各结点的结点表,从而在保持和改善广播操作的时间性能的同时有效地降低了系统资源的消耗以及路由算法的设计复杂度。

在本文的讨论中,我们假设在一个单位时间(本文中又称为一步)内,一个结点只能向一个相邻结点传送数据,并且认为可以完成传送。另外,该模型允许一个结点同时接收多个结点传来的数据,随机选择其中之

*)本课题得到国家自然科学基金资助。胡宁宁 硕士生,主要研究领域为分布与并行计算机系统、网络系统。陈道蓄 教授,主要研究领域为分布与并行计算机系统。

一进行处理。

记 M 为某种广播模型(可为 cl(classical list)或 il(intelligent list)),G 为由实际网络抽象所得的图,s 为广播源点,σ 为在 M 中所使用的具体广播策略,例如对 ILBM 而言就是指结点算法。这里给出本文使用的符号与术语:①$B_M(G,s,\sigma)$表示当广播源为 s 时,使用 M 模型的广播策略 σ,在拓扑结构 G 中完成广播(所有结点都获得广播数据)所需的步数;当没明确指出 s 和 σ 时,它们可取任意值。②$t_M(A,s,\sigma)$表示以 s 为源点,在模型 M 中使用广播策略 σ 时结点 A 第一次获得广播数据所需时间(步数)。③X′表示对 X 的二进制取反(其中 X 可为一位或多位二进制数)。④一级路径:若路径$(v_0,v_1,\cdots,v_n)$中任一 v_{i+1}都为 $v_i(0\leqslant i<n)$进行广播时由结点算法第一步所选择的后继结点,则称该路径为一条以 v_0为起点的一级路径。

本文首先简单介绍了 CCC 结构的特点及相应概念,再介绍传统广播模型与全局表广播模型在 CCC 中的应用情况,最后给出了智能表广播模型的构造方法及其在 CCC 中的性能分析。

2 立方体连接环(CCC)简介

在 CCC 结构中,所有处理器元素都用三个端口相互连接在一起。任一链接都可用于数据的双向传输。CCC 图一般记为 CCC(b,k),表示它包含 2^k 个环,每个环都有 h 个结点,其中 $b\geqslant k$。实际上 CCC 是超立方体的一种改型,旨在减少每个结点的链接数目,改善系统性能。将超立方体中每个结点以一组结点(b 个)代替,这组结点组成环,同时将超立方体中连接到同一结点的 k 条链接分配给环中的 h 个结点,每个结点最多一条(因而一般要求 $h\geqslant k$)。易见,CCC(h,k)中包含 $h*2^h$ 个结点,每个结点都被赋予一个地址对(c,p),其中 c 指出该结点属于哪一个环,而 p 则指出该结点在环中的位置,易见$0\leqslant c\leqslant 2^k$,$0\leqslant p\leqslant b-1$。

一般记 PE 的3个链接为 F(向前),B(向后),L(向侧)。CCC 图中的所有结点按以下规则连接:

①:PE(c,p)的 F 与 PE(c,(p−1)mod h)的 B 相连;

②:PE(c,p)的 B 与 PE(c,(p−1)mod h)的 F 相连;

③当$0\leqslant p<k$ 时,PE(c,p)的 L 与 PE$(c+\alpha*2^p,p)$的 L 相连,其中 $\alpha=1-2bit_p(c)$($bit_p(c)$表示当 c 用二进制表示时的第 p 位),即若 c 的二进制表示为 $c_{k-1}\cdots c_0$,则 PE$(c_{k-1}\cdots c_0,p)$的 L 与 PE$(c_{k-1}\cdots\overline{c_p}\cdots c_0,p)$的 L 相连;当 $k\leqslant p<h$ 时,PE(c,p)的 L 不加利用。

例如,CCC(4,3)如图1。

3 一般广播模型

对于紧耦合多处理器系统中的广播操作,网络中结点的邻点选取策略(即当结点获取到广播数据时,如何选取其相邻点进行转发)是决定广播时间的重要因素之一。为使广播时间尽可能短,对同一结点而言,当源点不同时其邻点选取策略一般也应不同。传统广播模型 CLBM(见图2a)利用在同一结点中存放多个表的方法来适应源点不同的情况,最坏的情况下,在一个结点数为 n 的网络中,某些结点需存放 n−1 张表,这样显然可以使广播时间在任何情况下都可达到最小,但它的一个明显的缺点就是要占用过多的存储器,在某些情况下(如 VLSI 中每个处理器所拥有的存储器相当有限)根本无法实现。另一个最致命的缺点就是当结点数达到一定程度后,为每个结点都找到这样的表简直是不可能的。

图1 CCC(4,3)

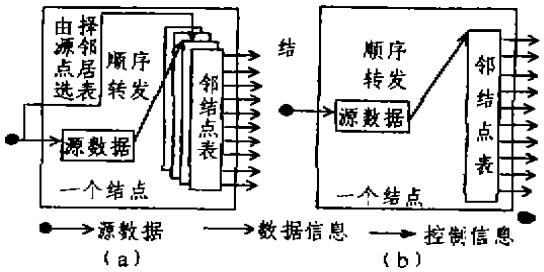


图2 两种广播模型(a)CLBM (b)ULBM

文[2]对这种模型进行了改进,提出了全局表模型 ULBM(见图2b),即在每一结点中仅维护一张邻结点表,只要接收到数据,就按预定顺序使用该邻结点表转发,而不管广播源点是那个结点。同时,ULBM 通过精心设计每一个邻结点表而使广播较好地适应不同源点的情况。文[2]中提出了 ULBM 的两种子模型:非适应

性全局表模型(NAULBM)与适应性全局表模型(AULBM)。在NAULBM中,当一个结点接收到一消息后,就先向邻结点表第一项所指向的结点转发,依次为第二项、第三项,直到所有相邻点都被转发过。在AULBM中,其转发过程与前者基本相同,唯一区别就在于不再向该消息的发送结点转发。

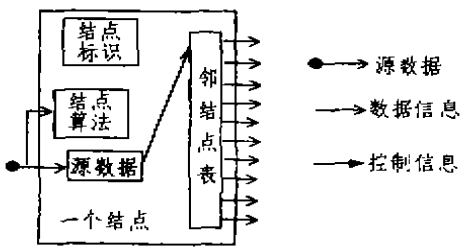
ULBM的突出优点是只使用一张邻结点表和简单的转发算法,而且对某些特定的拓扑结构,ULBM可通过对邻结点表的精心设计而达到与CLBM相同的时间效果。但同时这种模型也有其局限性:①为能较快地完成广播,通常不能对所有结点按统一规律构造邻结点表,因而邻结点表中各结点顺序往往在空间上不一致,而恰恰是这种不一致,造成了较大的实现开销与设计的困难。②在选取相邻结点时,ULBM只是简单地使用顺序选取算法,路由选择缺乏较强的自适应能力。从文[2]中可知,ULBM在CCC中的设计将是非常繁琐的,其执行效率的证明也将比较困难。

4 智能表广播模型及其在CCC中的应用

针对CCC结构的特点,本文提出一种新的广播模型—智能表广播模型ILBM(见图3a),利用算法本身的智能性来充分利用CCC结构的特点,从而达到方便设计、减小系统开销之目的。

4.1 智能表广播模型ILBM

在ILBM中,每一结点都被赋予一个邻结点表与



(a)

```
Algorithm selection_algorithm;{prototype}
begin
    srcLinkId=getSrcLinkId();
    nodeId=getNodeId();
    i=getFirstRetransmitLink(srcLinkId,nodeId);
    send message to List[i];
    while (retransmission not finish)
    begin
        i=get Next Link(srcLinkId, nodeId, i);
        send message to List[i];
    end;
end.
```

(b)

图3 (a)ILBM;(b)结点算法的伪代码

一个转发算法,邻结点表的项就对应于结点的邻结点,而转发算法(下文称这种转发算法为结点算法,图3b是其伪代码)则用来决定转发顺序。当一结点获得广播数据时,就运行结点算法,决定邻结点表的哪一项首先转发,哪一项次之,等等。在ILBM中,一个具体的网络只有一个结点算法,即网络中所有的结点都使用相同的结点算法。同时,邻结点表也因此而统一了,这种统一并不是指它们完全相同,而是指一结点的邻结点与其在表中的位置有相同的映射。

由于算法属于软件范畴,可以在运行过程中根据不同的情况灵活处理,即同一算法在不同情况下运行效果可以不同,从而以静态形式上的一致实现运行效果的不一致,这样就较好地避免了ULBM中针对不同结点设计不同邻结点表所带来的困难。另外,算法的设计相对于邻结点表的设计具有较大的灵活性,这有助于简化ILBM的设计过程。正是上述唯一的结点算法与统一的邻结点表使得ILBM可以用较简单的设计达到ULBM相同甚至更好的效果,而且,作为软件的算法本身所具有的灵活性也使得该模型更容易利用特定拓扑结构内在的性质而达到更好的效果。

综上,与CLBM、ULBM相比,ILBM具有低内存消耗、强适应性以及易于设计等优点。

4.2 ILBM在CCC图中的应用

本节中我们证明用ILBM设计的算法可在(2k-1+⌈(h-1)/2⌉)到(2k-1-2⌈(h-1)/2⌉)步内完成对CCC(h,k)的广播,这与文[12]中指出的CCC广播性能是一致的。在CCC结构中进行广播的ILBM模型算法CCCBrd如下:

算法CCCBrd;

```
/* 这是一个使用ILBM在CCC中进行广播的算法。在这个
算法中,我们使用下面的函数来发送广播消息:send(X,cnt1,
cnt2,flag):通过链接X(F、B或L)发送消息,但当cnt1<=0
and cnt2<=0,时,不进行广播,cnt1,cnt2,flag是我们在广播
消息中增加的一些字段;
cnt1用来当一个环完成其广播时让算法停止;
cnt2用来在CCC中的所有环已完成广播时停止算法;
flag代表当前结点是否在源点中;*/
{
if(这是原结点)
/*对于源环,我们在其中使用某一不同的广播策略*/
{
    i=getCycleId();/*获得源点在源环中的位置*/
    mid=(2*k-1-h)/2;
    if(i>mid)
    /*让型如(c_{k-1}...c_0,k-1)的结点尽快获得原消息*/
      {send(F,⌊(h-1)/2⌋,-1,1);
                  send (B,⌈(h-1)/2⌉,-1,1);}
    else
      {send (B,⌊(h-1)/2⌋,-1,1);
           send(F,⌈(h-1)/2⌉,-1,1);}
       send(L,-1,-1,0);
}
else if (getFrom_F(cnt1,cnt2,flag))
     /*这个结点从其F链接获得原消息*/
{
     if(flag)/*若当前结点在源环中*/
       {send(B,cnt1-1,cnt2,flag);
        send(L,-1,k,0);}
     else (send(L,-1,cnt2-1,flag);
          send(B,cnt1-1,cnt2,flag);}
}
```

```
    else if(getFrom_B(cnt1,cnt2,flag))
        /* 这个结点从其 B 链接获得源消息 */
    {
        if(flag)/* 若当前结点在源环中 */
          {send(F,cnt1-1,cnt2,flag);
           send(L,-1,k,0);}
        else{send(L,-1,cnt2-1,flag);
             send(F,cnt1-1,cnt2,flag);}
    }
    else if(getFrom_L(cnt1,cnt2,flag))
        /* 这个结点从其 L 链接获得源消息 */
    {
        send(L,-1,cnt2-1,flag);
        send(B,⌊(h-1)/2⌋,-1,flag);
        send(F,⌈(h-1)/2⌉,-1,flag);
    }
}
```

该算法的基本思想是：在固定的时间内让尽量多的结点环获得数据，即每一结点环获得数据后都应尽量快地将其转发到别的结点环中，从而总是先向 L 链所指向的邻结点转发。

引理1 $t_{ll}((c_{k-1}\cdots c_1)\cdots c_0,i),(c_{k-1}\cdots c_0,k-1),CCCBrd)=2(k-i)-1$，其中$0\leqslant i\leqslant k-1$。

证明：从源点$(c_{k-1}\cdots c_0,k-1)$到$((c_{k-1}\cdots c_1)\cdots c_0,i)$的一级路径如图4。其中1型箭头表示沿 L 链的转发，在 CCCBrd 中需用1步完成；2型箭头表示沿 B 链的转发，在 CCCBrd 中需用2步完成；3型箭头表示先沿 L 链再沿 B 链的转发，在 CCCBrd 中需用3步完成。易于计算该一级路径共需$2(k-i)-1$步转发，即$t_{ll}(((c_{k-1}\cdots c_1)\cdots c_0,i),CCCBrd)=2(k-i)-1(0\leqslant i\leqslant k-1)$。

引理2 $t_{ll}((c_{k-1}\cdots c_{i[m]}\cdots c_{i[m-1]}\cdots c_{i[0]}\cdots c_0,i[0]),(c_{k-1}\cdots c_0,k-1),CCCBrd)\leqslant t_{ll}(((c_{k-1}\cdots c_{i[0]})\cdots c_0,i[0]),(c_{k-1}\cdots c_0,k-1),CCCBrd)$，其中$0\leqslant i[0]<i[1]<\cdots<i[m-1]<i[m]\leqslant k-1$。

证明：(略)

由引理1与引理2可得：

推论1 $t_{ll}((c_{k-1}\cdots c_{i[m]}\cdots c_{i[m-1]}\cdots c_{i[0]}\cdots c_0,i[0]),(c_{k-1}\cdots c_0,k-1),CCCBrd)\leqslant 2(k-i[0])-1$，其中$0\leqslant i[0]\leqslant k-1$。

引理3 若广播源为$(c_{k-1}\cdots c_0,k-1)$，则整个 CCC 可在$(2k-1+\lceil(h-1)/2\rceil)$步内完成广播。

证明：对 CCC 中的任一结点环 $b_{k-1}\cdots b_0$，记 $i=\min\{j|b_j=c_j,0\leqslant j\leqslant k-1\}$，由推论1可知 $t_{ll}((b_{k-1}\cdots b_0,i),(c_{k-1}\cdots c_0,k-1),CCCBrd)=t_{ll}(b_{k-1},\cdots\cdots b_{i+1}c_i\cdots c_0,i),(c_{k-1}\cdots c_0,k-1),CCCBrd)=2(k-i)-1$，在结点环 $b_{k-1}\cdots b_0$中有一个结点可在$2(k-i)-1$步内得到数据，由算法 CCCBrd 可知，再经$2i$步，结点$(b_{k-1}\cdots b_0,j)$，$0\leqslant j\leqslant 3i$也都将得到数据，此时在该环内广播数据不会再先走 L 链，即以后的结点都以 B 或 F 为优先转发链，从而使结点环再用$\lceil(h-1)/2\rceil$步即可完成本环的广播，即共需$2(k-i)-1+2i+\lceil(h-1)/2\rceil=2k-1+\lceil(h-1)/2\rceil$步完成该环的广播。

由结点环 $b_{k-1}\cdots b_0$的任意性可知，当广播源为$(c_{k-1}\cdots c_0,k-1)$，CCC 可在$2k-1+\lceil(h-1)/2\rceil$步内完成所有结点环的广播。□

若广播源点不在$(c_{k-1}\cdots c_0,k-1)$，而在该环的别处，则我们就优先在该源环内广播(见 CCCBrd 中对源环的处理代码)，最多经$\lceil(h-1)/2\rceil$步可达$(c_{k-1}\cdots c_0,k-1)$，再利用引理3可知整个 CCC 可在$2k-1+2\lceil(h-1)/2\rceil$步内完成广播，从而有定理1(详细证明从略)：

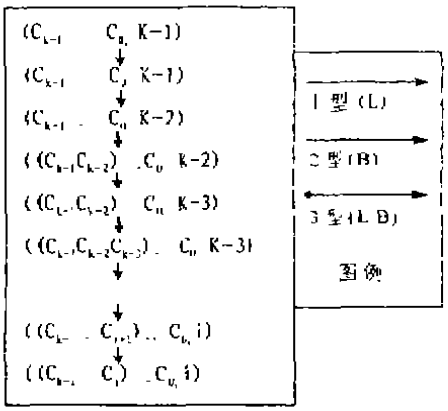


图4 从$(c_{k-1}\cdots c_0,k-1)$到$((c_{k-1}\cdots c_i)\cdots c_0,i)$的一级路径

定理1(CCC) $B_{ll}(CCC(h,k))\leqslant B_{ll}(CCC(h,k),CCCBrd)=2k-1+2\lceil(h-1)/2\rceil$。

结论 事实上，ILBM 并不只适用于 CCC 这一种结构，其设计思想可应用其它所有具有明显规律的紧耦合系统。只要结点算法能充分利用相应拓扑结构的特点，一般就能方便地设计出相应的算法，从理论上讲，ILBM 对任何拓扑结构都可达到 CLBM 相同的效果(仅考虑时间性能)，因为我们可以用复杂的结点算法来达到使用大量存储空间才能达到的效果。当网络的拓扑结构规律性不强时，结点算法一般会较复杂，因而实用性不高；但当网络拓扑结构规律性很强，则具有较高的灵活性及实用性。

应用 ILBM 的关键在于结点算法的设计，应尽量避免在一级路径中某结点同时收到多条消息的情况，因为本文已假设在广播过程中消息选择是随机的，而在多条消息中选择不同的消息进行处理会得到不同的一级路径，从而广播的结果也就不同，可以证明在 CCCBrd 中都不会出现这种情况。

参 考 文 献

1 Farley A M, Hedetniemi S T. Broadcasting in grid graphs. In: Congressus Numerantium XXI, Proc. of the 9th S-E Conf. Combinatorics. GraphTheory and Computing. 1978. 275～288

IP网络 QoS ATM Internet网 路由器 (18)

计算机科学2000Vol. 27№. 5

IP 网络中的 QoS 策略控制技术

The QoS Policy Control in IP Networks

76-80

洪佩琳 刘晓湘 杨凯峰 李津生

(中国科技大学电子工程与信息科学系 合肥 230027)

TP393

Abstract It is necessary to provide Quality of Service(QoS) with the continuous growing of the multimedia traffic in the Internet. In this paper, we analyze several QoS models, describe that why the QoS should be policy control in the Internet, present the detail realization scheme of QoS policy control.

Keywords QoS, Policy control, Differentiated service

一、引言

随着 Internet 规模的不断扩大，IP 网上的实时业务量也在不断增长，由于实时业务对网络传输的延时、延时抖动等特性较为敏感，当突发性高的 FTP 或含有图像文件的 HTTP 等业务量在网络上传输时，实时业务会受到影响，因此应在 IP 网络上导入 QoS 技术，以确保实时业务的通信质量。

最初，QoS 是与 ATM 技术紧密相连的，随着 IP 网络上实时业务的增多，人们开始研究 IP 网络上的 QoS 技术。早期的许多工作都源于资源预留协议(RSVP, Resource Reservation Protocol)，但是由于 RSVP 的实现比较复杂，用户在通信前必须通过信令在收发两端沿途各转发节点上进行资源预留，也就是说，沿途各节点上的路由器都要支持 RSVP 协议。这种实现方法既加重了路由器的处理负担，又会增加网络的数据量，因此没能得到普及。

近两年来，随着 IP 电话、可视电话、远程教学等多媒体实时业务的开发和普及，对 IP 网上的服务质量的要求更加迫切，并相继提出了各种与 QoS 有关的技术，IETF 还成立了区分服务(Diff-serv)工作组，专门从事区分服务的标准化工作。一些大的通信厂商也联合成立了 QoS 论坛，协商各种 QoS 技术标准的实施方案。

二、多种多样的 QoS 技术

1. 按层次对 QoS 技术分类

广义地讲 QoS 技术的种类有很多，由于处理的层次及采用的技术不同，其效果也不一样。如图1所示，从物理层到应用层，QoS 利用了各式各样的技术，每一层都试图通过对业务流的适当控制来改善或保证通信质量。

物理层中按业务量分配固定带宽的 TDM(时分复用)，虽然能够严格保证 QoS，但往往不称其为 QoS 技术；在 ATM 网络中，各种业务流可分成四种不同的业务类型，终端在通信前首先要建立连接，网络根据用户

2 Diks K, Pele A. Broadcasting with Universal Lists. Networks, 1996, 27(3): 183～196

3 Koren I. A reconfigurable and fault-tolerant VLSI multiprocessor array. In: Proc. 8th Int. Symp. Comput. Architecture. Minneaplois. MN, May 1981. 425～442

4 Chung F R K, et al. DIOGENES: A methodology for designing fault-tolerant VLSI processor arravs. In: Proc. 13th Fault-Tolerant Compur. Symp. June 1983. 26～32

5 Samu M, Stefanelli R. Reconfigurable architectures for VLSI implementation. In: Proc. Nar. Compur. Conf. May 1983. 565～577

6 Hedlund K S, Snyder L. Water-scale integration of conrigurable highly parallel processor. In: Proc. Int. Parallel Processing Conf. Aug. 1982. 262～264

7 Snyder L. Introduction ro rhe configurable, highly parallel computer. Computer, 1982, 15(1): 47～56

8 Pease M C. The indirect binary n-cube microprocessor array. IEEE Trans. on Comput, 1977, C-26(5): 458～473

9 Atallah M J, Kosaraju S R. A generalized dictionary machine for VLSI. IEEE Trans. on Comput, 1985, C-34(2): 151～155

10 Stone H S. Parallel processing with the perfect shuffle IEEE Trans. on Comput, 1971, C-20(2): 153～161

11 Preparata E P, Vuillemin J. The cube-connected cycles: A versatile network for parallel computation. Conunnun. ACM, 1981, 24(5): 300～309

12 Nian-Feng Tzeng. A cube-connected cycles architecture with High reliability and Improved Performance. IEEE Trans. on Comput. , 1993, 42(2): 246～253