第6卷第4期 软 件 学 报 Vol. 6, No. 4
1995年4月 JOURNAL OF SOFTWARE Apr. 1995

超立方体上基于缓冲机制的无死锁路径算法*

周建强 姚学军 谢 立

(南京大学计算机科学系,南京 210093)

摘要 本文研究了超立方体上基于单缓冲和双缓冲技术的无死锁受限条件,提出了相应的无死锁路径算法.性能分析表明,路径算法的效率和算法的自适应能力及算法的复杂性相关.

关键词 超立方体,路径算法,死锁,缓冲,自适应.

超立方体是无共享存贮耦合的多处理器系统采用的一类重要互连拓扑结构.信件传递是诸处理器进行通信的唯一途径.因此,在基于这类耦合模式的多处理器系统中,路径算法占有极其重要的地位.

路径算法必须考虑安全性和有效性.前者主要涉及算法的无死锁性.死锁是指系统在某一时刻存在若干信件因彼此等待网络信道资源而永远无法到达终点的情形.

防止死锁一般有两种方法.一是给定一组无死锁路径算法满足的受限条件.例如,超立方体上的固定式路径算法预先规定了通信信道的一个次序[1],信件必须按序地占有信道资源以破坏死锁的循环等待条件.这类算法固然形式简洁优美,但是,它所遵循的有序受限条件,使得系统内任意两点间只有一条固定路径可供利用.文献[2]提出的一类基于弱化有序受限条件的无死锁路径算法可以在两结点间提供多条不同路径.但是,这类算法显示的非均衡特性又限制了算法性能的进一步提高.防止死锁的另一类方法是利用缓冲技术[3-5].例如,在每个处理器结点上开辟若干信件缓冲,并预先规定诸缓冲间的一个次序,信件按序地占有缓冲资源而实现在诸结点间的传递.但是,基于缓冲技术的无死锁路径算法必须付出用于缓冲管理的开销.

结合无死锁受限条件和缓冲技术,本文研究了超立方体上的无死锁路径算法.本文第1节定义超立方体的拓扑结构和死锁的概念,第2节提出两组最小无死锁受限条件并描述相应的路径算法,第3节将双缓冲技术应用于无死锁受限条件,提出两组基于双缓冲的无死锁受限条件和路径算法,并且证明这类算法可以利用超立方体上最大数目的相异路径,第4节是性能分析,最后为结束语.

* 本文1993-05-26收到,1993-10-04定稿

作者**周建强**,1961年生,讲师,主要研究领域为分布并行处理,操作系统国际化.**姚学军**,1968年生,硕士,主要研究领域为分布计算系统.**谢立**,1942年生,教授,南京大学副校长,主要研究领域为分布并行处理,操作系统,人工智能.

本文通讯联系人:谢立,南京 210093,南京大学计算机科学系

1 表示和定义

下文涉及图论中的一些术语和概念,读者可参阅文献[6]. 设 $G(V,E)$是一个图,$V(G)=V$ 和 $E(G)=E$ 分别表示 G 的结点集和边集,如果 $u,v\in V$,且$(u,v)\in E$,则称 u 和 v 是相邻的. 在下文中,结点又代表处理器,而边表示连接相邻处理器的两个相向通信信道.

用 $b=b(n)b(n-1)\cdots b(1)$表示一个 n 位的二进制数,$b(i)\in\{0,1\}$,$i=1,2,\cdots,n$. 用$\overline{b}(i)$表示 $b(i)$的相反数($\overline{0}=1,\overline{1}=0$). $\overline{b}=\overline{b}(n)\overline{b}(n-1)\cdots\overline{b}(1)$. $\|b\|$ 等于 b 中二进制位为 1 的个数. $\oplus$表示异或运算符.

定义 1.1. n 维超立方体 $Q_n(V,E)$是一个图,其中 V 含 2^n 个结点,它们分别用 0 至 2^n-1的二进制数唯一地标识. 设 $u,v\in V$,则若$(u,v)\in E$,当且仅当 $\|u\oplus v\|=1$.

设 u 和 v 是 Q_n 上的两个相邻结点. 若 $u\neq v$,则称 u 和 v 是相向结点. 用 $c=(u,v)$表示 u 至 v 的信道(其相向信道为(v,u)),其中 u 和 v 分别称为 c 的始点和末点. 在信道的始点有一个或若干信件缓冲(可容纳一最长信件),它们分别用 1,2,…顺序编号. 若有 i 使 $u(i)\neq v(i)$,则称 c 为 i 维信道. 若又有 $u(i)=0$,则称 c 为正信道,否则称 c 为负信道. 信道 c 和其始点的缓冲 i 构成一逻辑信道,简称为 i 类信道. 信件在逻辑信道上的传递实现诸处理器间的通信,它满足以下性质.

(1)等待性. 当逻辑信道被一信件占用后(信件占有了该逻辑信道相应的缓冲),其它欲使用该逻辑信道的信件则必须等待.

(2)非剥夺性. 一信件占用了逻辑信道后,其它欲申请该信道的信件不能抢占使用.

(3)非释放性. 一信件欲申请下一逻辑信道而得不到满足时,不释放已占用的逻辑信道.

设 c_i 和 c_j 是 Q_n 上的两个逻辑信道,$c_i=(n_1,n_2)$和 $c_j=(n_2,n_3)$,它们分别被信件 M_i 和 M_j 占用. 如果信件 M_i 欲申请使用 c_j 时,则 M_i 处于占用 c_i 且等待 c_j 的状态,简称 c_i 等待 c_j. 显然,若有 c_i 等待 c_j,则必有 $n_1\neq n_3$.

定义 1.2. 如果系统在某一时刻存在一个循环等待的逻辑信道链 $c_0,c_1,\cdots,c_{n-1}$,其中 c_i 等待 $c_{(i+1)\bmod n}$,$i=0,1,\cdots n-1$,则称系统发生了死锁.

下述引理扩展了文献[2]的结论.

引理 1.1. 对于超立方体 Q_n 上的任意一个逻辑信道的循环等待链,必存在相同数目的同维正信道和负信道.

2 单缓冲无死锁路径算法

本节讨论超立方体上基于单缓冲(信道始点只有一个缓冲,此时逻辑信道又可简称为信道)的无死锁受限条件及其相应的路径算法.

2.1 受限条件

定义 2.1. 设 c_i 和 c_j 为 Q_n 上的两个信道,若 c_i 等待 c_j,则受限条件 B_{11}定义如下:

(1)c_i 为正信道;或

(2)若 c_i 为负信道,则 c_j 为负信道.

定理 2.1. 满足受限条件 B_{11}的路径算法是无死锁的.

证明:若满足受限条件 B_{11} 的路径算法是有死锁的,则某一时刻系统中有一信道的循环等待链 $c_0,c_1,\cdots,c_{n-1}$,其中 c_i 等待 $c_{(i+1)\bmod n}$,$i=0,1,\cdots,n-1$. 根据引理 1.1,上述诸信道中必有一负信道,不妨设 c_0 为负信道,根据 B_{11},c_1 至 c_{n-1} 必都为负信道,这和引理 1.1 矛盾,故定理成立.

用 $\Sigma_{B,n}$ 表示 Q_n 上满足受限条件 B 的所有路径的集合. 称 B 是无死锁的,若信件在 $\Sigma_{B,n}$ 规定的路径上传递时不可能产生信道的循环等待链. 称 B 为最小无死锁受限条件,若 B 是无死锁的且在 $\Sigma_{B,n}$ 中增加任一新路径(其长度大于 1)必产生死锁.

定理 2.2. B_{11} 是最小无死锁受限条件.

证明:在 $\Sigma_{B11,n}$ 中加入一新路径 P,因此,P 中必有子路径 $\{n_{i-1},n_i,n_{i+1}\}$,使得 $c_{i-1}=(n_{i-1},n_i)$ 等待 $c_i=(n_i,n_{i+1})$,且不满足 B_{11}. 根据 Q_n 的拓扑性质,构作 Q_n 上的二维超立方子体 $Q_n[n_{i-1},n_i,n_{i+1}]$,它包含结点 n_{i-1}、n_i 和 n_{i+1},且同构于 Q_2. 图 1 显示满足 B_{11} 的 6 条路径,而在其中增加任一新路径必然可构造一信道的循环等待链. 根据同构关系,在 $Q_n[n_{i-1},n_i,n_{i+1}]$ 上增加子路径 $\{n_{i-1},n_i,n_{i+1}\}$ 时可以对应地构造其上的一个信道的循环等待链. 因此定理成立.

P_1:{00,01,11}　　P_2:{00,10,11}
P_3:{11,10,00}　　P_4:{11,01,00}
P_5:{01,11,10}　　P_6:{10,11,01}

图 1

设 $u,v\in V(Q_n)$,u 至 v 的两条路径称为相异路径,若它们除源点和终点外不含有其它相同结点. 设 $\|u\oplus v\|=k$. 记 $diff(u,v)=\{b_1,b_2,\cdots b_k\}$,其中 $b_1<b_2<\cdots<b_k$,且 $u(b_i)\neq v(b_i)$,$i=1,2,\cdots,k$. 记 $diff^+(u,v)=\{a_1,a_2,\cdots,a_k\}$,和 $diff^-(u,v)=\{b_1,b_2,\cdots,b_{k2}\}$,其中 $u(a_i)\neq v(a_i)$ 且 $u(a_i)=0$,$i=1,2,\cdots,k_1$ 和 $u(b_i)\neq v(b_i)$ 且 $u(b_i)=1$,$i=1,2,\cdots k_2$. 显然 $k_1+k_2=k$.

定理 2.3. 设 $u,v\in V(Q_n)$,$\|u\oplus v\|=k$,$diff^+(u,v)=\{a_1,a_2,\cdots,a_{k1}\}$,$diff^-(u,v)=\{b_1,b_2,\cdots b_{k2}\}$. 则 $\Sigma_{B11,n}$ 具有以下性质.

(1)若 $k_1=k$ 或 $k_2=k$,则 $\Sigma B_{11,n}$ 中存在 u 至 v 的 k 条相异路径. 否则

(2)$\Sigma B_{11,n}$ 中仅存在在 u 至 v 的一条相异路径. 并且 $\Sigma_{B11,n}$ 中所有 u 至 v 的不同路径都相交于一个共同点.

证明:若 $k_1=k$,则 $diff^-(u,v)$ 为空集,我们用如下的 k 个序列:

p_1: $a_1,a_2,\cdots\cdots,a_k$
p_2: $a_2,\cdots\cdots,a_k,a_1$
$\cdots\cdots$
p_k: $a_k,a_1,\cdots\cdots,a_{k-1}$

表示 Q_n 上的 k 条路径,其中序列中的每一个 a_i 表示信件使用相应结点的第 a_i 维信道而得到的一条路径. 显然,上述 k 条路径是相异的且满足 B_{11} 受限条件.

若 $k_2=k$,则证明同上.

若 $k_1\neq k$ 且 $k_2\neq k$. 设 P 是 $\Sigma_{B11,n}$ 中 u 至 v 的任一条路径,P 表示为序列 $\{c_1,c_2,\cdots,c_k\}$. 由受限条件可知,当 $i\leqslant k_1$ 时,$c_i\in diff^+(u,v)$;当 $i>k_1$ 时,有 $c_i\in diff^-(u,v)$. 以下容易推出

定理的第 2 个结论成立. 证毕.

文献[2]给出了另一组无死锁受限条件,本文记为 B_{12}.

定义 2.2. 设 c_i 和 c_j 分别是 Q_n 上的 l 维和 m 维信道,若 c_i 等待 c_j,则受限条件 B_{12}定义如下.

(1)c_i 为正信道,或

(2)若 c_i 为负信道,则 $m>l$.

下面的两个定理介绍了文献[2]的结果.

定理 2.4. B_{12}是最小无死锁受限条件.

定理 2.5. 设 $u,v\in V(Q_n)$,$diff(u,v)=\{b_1,b_2,\cdots,b_k\}$. 若 $u(b_1)=0$,则 $\Sigma_{B12,n}$中存在 u 至 v 的 k 条相异路径. 否则,$\Sigma_{B12,n}$中仅存在 u 至 v 的一条相异路径.

2.2　路径算法

超立方体中每一个处理器上有一个路径选择器负责结点间信件的传递. 它由信件转发进程和用户接口进程两部分构成(图 2). 前者从输入端发送信件或上交用户接口进程. 后者接纳信件转发进程的信件并移交给上层的用户进程,或者接纳由上层用户进程下达的信件,并由路径算法选择一输出端发送. 本部分仅讨论基于单缓冲的路径算法,因此,每一个输出端配备一个信件缓冲,而输入端有无缓冲则视具体实现的不同而不同.

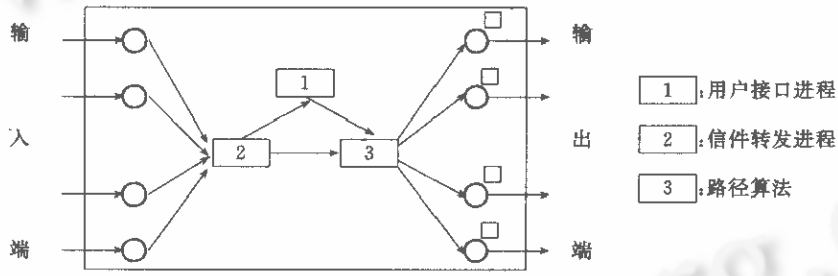

图2

图 3 描述了基于受限条件 B_{11}的路径算法,其基本思想是:信件传递过程中首先使用正信道,当无正信道时,再使用负信道.

R_{11}:当从输入端接收到信件 M 或用户接口进程要求发送上层用户进程的信件 M 时,启动本算法工作,设 I 是本地结点号,D 是信件 M 的目的结点号.

(1)$x=I\oplus D$
(2)if ($x==0$) Then
(3)　信件 M 上交用户接口进程,算法结束
　fi
(4)if (存在 i,有 $x(i)==1$ 且 $I(i)==0$) then
(5)　任选一个满足条件的 i. 信件 M 沿 I 的 i 维信道发出
　else
(6)任选一个 I,有 $x(i)=1$. 信件 M 沿 I 的 i 维信道发出
(7)*fi*
(8)算法结束.

图 3

图 3 中的第 5 步和第 6 步反映了算法具有一定的自适应能力,即算法可以在满足条件

的诸信道中根据一定的自适应策略选择其中一个信道使用[7]. 基于B_{12}的无死锁路径算法R_{12},请读者参阅文献[2]. 比较而言,这类算法的自适应能力较弱.

3 基于双缓冲的无死锁路径算法

上文提出了超立方体上基于单缓冲的两组最小无死锁受限条件. 这类算法显示了非对称或非均衡的特点,即Q_n上不可能存在两对相向结点,它们之间同时存在多于一条的相异路径. 下文讨论基于双缓冲(信道始点有两个缓冲)的无死锁受限条件和相应的路径算法.

3.1 受限条件

定义 3.1. 设c_i和c_j是Q_n上的两个逻辑信道,若c_i等待c_j,则受限条件B_{21}定义如下:

(1)c_i是1类信道,c_j为2类信道;

(2)若c_i和c_j为1类信道,则c_i为正信道,或若c_i为负信道,则c_j也为负信道;

(3)若c_i和c_j为2类信道,则c_i为正信道,或若c_i为负信道,则c_j也为负信道.

定理 3.1. B_{21}是无死锁受限条件.

证明:我们把信道分成两类资源,根据B_{11},同类资源不可能构成循环等待链. 又不同类资源中只有1类信道等待2类信道,因此也不可能构成循环等待链. 故定理成立.

定理 3.2. 设$u,v\in V(Q_n)$,若$\|u\oplus v\|=k$,则$\Sigma_{B21,n}$中存在u至v的k条相异路径.

证明:设$diff^+(u,v)=\{a_1,a_2,\cdots,a_{k1}\}$,$diff^-(u,v)=\{b_1,b_2,\cdots b_{k2}\}$,其中,$k_1+k_2=k$,以下假定$k_1\neq k$且$k_2\neq k$,否则定理已经成立. 构作如下的$k$条相异路径:

$$p_1:b_1,b_2,\cdots,b_{k2},a_1,a_2,\cdots,a_{k1};$$

$$P_2:b_2,b_3,\cdots,b_{k2},a_1,\cdots,a_{k1},b_1;$$

$$\cdots\cdots$$

$$p_{k2+1}:a_1,a_2,\cdots,a_{k1},b_1,\cdots,b_{k2};$$

$$\cdots\cdots$$

$$p_k:a_k,b_1,\cdots,b_{k2},a_1,\cdots,a_{k-1}.$$

对于上述的任一路径,我们首先使用1类信道,当遇b_{k2}维信道后,我们转而使用2类信道. 因此,上述路径满足受限条件B_{21}. 证毕.

定义 3.2. 设c_i和c_j是Q_n上的l维和m维逻辑信道,若c_i等待c_j,则受限条件B_{22}定义如下.

(1)c_i为1类信道,c_j为2类信道;

(2)若c_i和c_j为1类信道,则c_i为正信道;或若c_i为负信道,则$m>l$;

(3)若c_i和c_j为2类信道,则c_i为负信道;或若c_i为正信道,则$m>l$.

对于受限条件B_{22},我们有类似的结论.

定理 3.3. B_{22}是无死锁受限条件. 设$u,v\in V(Q_n)$,若$\|u\oplus v\|=k$,则$\Sigma B_{22,n}$中存在u至v的k条相异路径.

3.2 路径算法

将双缓冲技术应用于B_{11}和B_{12},本文分别获得了受限条件B_{21}和B_{22}. 事实上,读者可以给出更为一般的双缓冲受限条件,其中只要保证1类信道和2类信道等待的有序性和同类

信道的无死锁性.本文仅给出基于 B_{21}的路径算法(图 4).

```
R21:设 I 为本地结点号,D 是信件 M 的目的结点号,C 为信件 M 占用的逻辑信道.
(1)x=I⊕D
(2)if (x==0) then
(3)    信件 M 上交用户接口进程,算法结束
   fi
(4)if (c 是 1 类信道) then
(5)    任选一个 i,有 x(i)=1
(6)    if (c 为正信道,或 I(i)=1) then
(7)        信件 M 沿 i 维 1 类信道发出
       else
(8)        信件 M 沿 i 维 2 类信道发出
       fi
(9)elif (存在 i,有 x(i)=1 并且 I(i)=0) then
(10)   任选一满足条件的 i,信件 M 沿 i 维 2 类信道发出
   else
(11)   任选一 i,有 x(i)=1,信件 M 沿 i 维 2 类信道发出
   fi
(12)算法结束.
```

图 4

注意到,算法在使用 1 类信道时具有完全的自适应能力(第 5 步),而在使用 2 类信道时仍具一定的自适应能力(第 10,11 步).因类同性,本文不再给出基于 B_{22}的路径算法 R_{22}.

4　性能分析

为了比较分析路径算法的性能,我们在多处理机 Transputer 上实现了 R_{11}、R_{12}、R_{21}和 R_{22}以及固定式路径算法 Fix 等五类算法.Transputer 由一个主处理机(Master)和 16 个从处理器(Slave)组成.Master 负责与宿主 PC 机(Host)接口及程序装载、网络拓扑重构等功能.16 个 T800 单片机通过高速通信链路互连构成 4 维超立方体,其中通信链路支持每秒 10 兆位、最高可达 20 兆位的全双工同步 DMA 通信.每个处理器上分别有 4 个输入端和 4 个输出端,每个输入端配备 1 个信件缓冲,而每个输出端配备 2 至 4 个信件缓冲.在双缓冲路径算法的实现中,上述缓冲分别分成 1 类或 2 类缓冲.当处理器单元接收到信件后,该信件或者上交用户进程(本地结点为信件的目的结点),或者由路径算法选择一输出端,将该信件排入此输出端的信件队列中等待发送.当输出端无空闲缓冲时,其它欲使用该输出端的信件则处于阻塞态.路径算法采用最短信件队列优先的自适应策略[7],即它总是选择满足无死锁条件且其输出端的信件队列为最短的信道发送信件.

本文的实验基于随机通信模型,每一处理器按信件发生频率定时地发送信件(信件的长度为 1k 字节),并且在单位时间(秒)内统计每一处理器接收到的信件数目.图 5 和图 6 显示了在各路径算法作用下,信件发生率和信件接收率两者之间的变化关系(其中 X 轴表示信件发生率,而 Y 轴表示信件发生率和信件接收率之差).

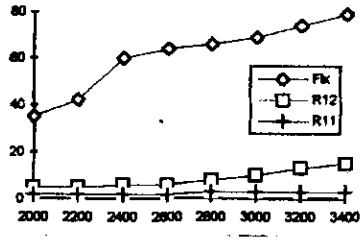

图 5 单缓冲无死锁路径算法

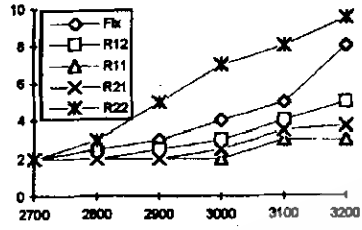

图 6 基于缓冲机制的无死锁路径算法

我们注意到本文的实验有两点不足：一是四维超立方体限制了试验规模，因此不能观察在高维超立方体上路径算法的性能的变化趋势；二是实际的通信模型往往显示了一类通信的局部化性质. 但是，从现有的试验结果来看，我们认为算法的性能和以下两个因素有关：

(1)算法的自适应能力. 注意 R_{11} 在路径选择的两个阶段具有较强的自适应能力，而 R_{12} 仅在路径选择的始点具有一定的自适应能力. 试验结果表明，路径算法的自适应能力越强，其性能就越好.

(2)算法的复杂性. 双缓冲类算法尽管可以最大限度地利用网络的通信能力，但是，图 6 的试验结果显示其性能并不优于单缓冲类算法. 我们认为，这是由于双缓冲类算法的复杂性导致了较大的运行开销(例如双缓冲机制的同步控制开销).

5 结束语

本文提出了二组分别基于单缓冲和双缓冲机制的无死锁受限条件，并描述了相应的路径算法. 从算法的形式看，单缓冲类较双缓冲类更具简洁性和优美性. 而从算法的性能来看，双缓冲类尽管充分地利用了网络的通信能力，但是，算法的复杂性导致的较高同步开销是影响这类算法可用性的主要障碍. 另外，路径算法的性能还和其自适应能力有关.

关于超立方体上当且仅当存在 B_{11} 和 B_{12} 两类最小无死锁受限条件以及路径算法的无死锁性和非均衡性两者之间的关系(在单缓冲情况下)，我们将另文论证.

参考文献

1 Sakai S, Yamagucji Y, Kodama Y *et al*. An architecture of dataflow single chip processor. Proc. International Symposiumon Computer Architecture, IEEE Computer Society, 1989. 46—53.

2 Qiang Li. Minimum deadlock—free message routing restrictions in binary hypercubes. Journal of Parallel and Distributed Computing, 1992(**15**):153—159.

3 Iannucci R A. Toward a dataflow/von neumann hybrid architecture. Proc. International Symposium of Computer Architecture, IEEE Computer Society, 1988. 131—140.

4 Sha L, Rajakumar, Ragunathan, Rehoczky J. Real—time scheduling support in futurebus+. Proc. Real—time System Symposium, 1990. 331—340.

5 Toda K, Nishida K, Uchibori Y *et al*. CODO: a multiprocessor architecture for sensor fusion. Proc. 5th International Symposium on Intelligent Control, 1990. 261—266.

6 Harary F. Graph theory. Addison—Wesley, Reading, MA, 1972.

7 Sivarama, P Dandamudi. A performance comparison of routing algorithms for hierarchical hypercube multicomputer networks. Proc. International Conference on Parallel Processing, 1990. I—281—285.

THE BUFFER—BASED DEADLOCK—FREE ROUTING ALGORITHMS IN HYPERCUBES

Zhou Jianqiang　Yao Xuejun　Xie Li

(*Department of Computer Science, Nanjing University, Nanjing* 210093)

Abstract　This paper presents a set of deadlock—free routing constraints in hypercubes based on the buffer technique, and describes their corresponding deadlock—free routing algorithms. The performance evaluation shows that the routing efficiency is dependent on both self—adaptability and complexity of the algorithms.

Key words　Hypercube, routing algorithm, deadlock, buffer, adaptability.