

路由算法

虚通道

DOR

互联网

19

70-73

基于 k-ary n-mesh 无虚通道的路由算法 PAM

Partial Adaptive Routing Algorithms for K-ary N-mesh without Virtual Channel

肖晓强 金士尧

(并行与分布处理国家重点实验室 长沙410073)

TP393

Abstract Most deadlock-free adaptive routing algorithms are based on virtual channels. Virtual channel increases the network throughput, and makes it easier to solve the deadlock problem. Nevertheless virtual channel makes the router more complex, and increases the setup latency. According to Chien's model, PAM routing algorithm, which has no virtual, deadlock-free and partial adaptive, is introduced based on k-ary n-mesh. Compared to the routing algorithms DOR (Dimension Ordered) and PAR (Planar Adaptive), the results show that PAM has the better performance.

Keywords Virtual channel, PAM routing algorithm, k-ary n-mesh, Router model, Setup latency

1 概述

大规模并行处理系统中各计算结点均依靠机间互联网进行通讯和协调。影响网络通讯性能的因素主要有拓扑结构、流控策略和路由算法等。目前针对互联网的研究工作主要集中于 k-ary n-cube 这样一类规则互联网。网络有 n 维, 每维上有 k 个结点, 每维上第 i ($i \in [0, k-2]$) 与第 i+1 个结点间有通道连接。当每维上第 k-1 与第 0 个结点间没有绕边 (wrap-around) 通道连接, 该网络就是 k-ary n-mesh^[1]。

现代的互联网多使用虫孔路由 WR (Wormhole Routing) 机制。使用该机制, 每个消息被分为若干个微片 (Flit)。其中第一个 (头) 微片包含有消息的路由信息, 由它负责开辟结点间的通讯通路, 其它微片以类似流水的方式紧跟头微片传输^[2]。微片的传输为原子过程, 即在每一步中, 微片要么在结点输入队列中等待, 要么经过该结点到达下一个结点的输入队列。采用 WR 机制, 每个结点所需的通讯缓冲小 (一个微片的大小), 并且, 当消息长度远远大于微片长度时, 消息的通讯延迟时间与通讯距离无关^[3]。这些都是 WR 机制备受关注的原因。对 WR 流控制的详细描述可参考文献 [4]。

采用 WR 机制带来的问题是: 微片占有资源等待, 网络很容易拥塞。为此, Dally 提出了虚通道技术来解决拥塞问题^[5]。虚通道技术使得网络吞吐率得到提

高, 且容易保证网络无死锁。因此, 使用 WR 机制的路由算法, 尤其是自适应路由算法一般都有虚通道。然而, 虚通道的引入也带来了实现上的复杂性等问题。本文基于 Chien 路由器模型, 提出了基于 k-ary n-mesh 上的一种无虚通道的、无死锁的部分自适应路由算法 PAM。

2 虚通道及 Chien 路由器模型

2.1 虚通道 VC

虚通道 VC (Virtual Channel) 是 Dally 提出的一种新型的流控制技术^[6]。它为每条物理通道增加缓冲区, 每个缓冲区对应一条虚通道, 所有虚通道以时分方式共享一条物理通道。这样使得活动的消息可以绕过阻塞的消息。因此, 许多采用 WR 机制的互联网都使用虚通道来达到提高网络吞吐率的目的。

虚通道的引入, 可以实现无死锁的自适应路由算法。例如 Duato 提出了路由算法无死锁的充分必要条件^[7], Glass 和 Ni 指出 k-ary n-cube 中, 当 $k > 3$ 时至少需要 2 条虚通道来保证无死锁^[8]。因此, 许多自适应路由算法都使用虚通道来保证无死锁。

然而, 虚通道的引入也带来一些问题: 1) 路由器对消息头的修改需要的开销比较大; 2) 虚通道需要增加缓冲区, 增加路由器交叉开关的规模, 需要更复杂的仲裁单元, 从而带来了实现上的复杂性; 3) 由于硬件规模的扩大, 使路由仲裁时间增加, 从而导致消息经过路由

肖晓强 博士生, 主要研究方向为计算机系统性能评价。金士尧 教授, 博士生导师, 主要从事高性能计算机体系结构和计算机仿真等方面的研究。

教授, 博士生导师, 主要从事高性能计算机体系结构和计算机

器的时间(路由器建立延迟)增加。

通过 A. A. Chien 的路由器性能模型^[10], 可以分析虚通道技术对路由器性能的影响。

2.2 Chien 模型

路由器由地址译码 AD、流控制 FC、交叉开关 CB、仲裁 ARB 和虚通道控制 VCC 等单元组成, 其基本结构如图1所示。

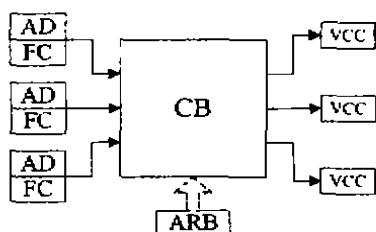


图1 路由器模型

消息的路由器建立延迟 T_{SETUP} 为消息经过各单元的延迟时间之和:

$$T_{\text{SETUP}} = T_{\text{AD}} + T_{\text{ARB}} + T_{\text{SEL}} + T_{\text{CB}} + T_{\text{VC}} \quad (1)$$

T_{SEL} 为自适应算法修改消息头需要的时间。

Chien 路由器模型根据实现各单元所需的门电路的级数来计算各单元上的延迟时间:

$$T_{\text{CB}} = C_0 + C_1 * \log_2 P \quad (P \text{ 为 CB 的端口数})$$

$$T_{\text{AD}} = C_2$$

$$T_{\text{ARB}} = C_3 + C_4 * \log_2 F \quad (F \text{ 为 CB 中可选路径数})$$

$$T_{\text{SEL}} = C_5 + C_6 * \log_2 F \quad (F \text{ 意义上同})$$

$$T_{\text{VC}} = C_7 + C_8 * \log_2 V \quad (V \text{ 为虚通道的个数})$$

Chien 根据 0.8 微米 CMOS 门电路技术, 统计出参数 $C_i (i \in [0, 8])$ 的值(见表1)。

表1 Chien 模型对应的 C_i 值

C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
0.4	0.6	2.7	0.6	0.6	1.4	0.6	1.24	0.6

2.3 虚通道的影响

为模拟网络的性能, 我们建立了 k -ary n -cube 互联网模拟器^[1]。模拟器中假定消息在通道上的传输时间很短, 微片经过一个结点到达下一个结点的传输时间由路由器建立延迟决定。

当不考虑虚通道对路由器建立延迟的影响时, 对 4-ary 2-mesh 的模拟结果见表2。结果表明: 使用虚通道可以有效地提高网络饱和请求率(网络允许的最大请求率), 从而提高网络吞吐量。

表2 4-ary 2-mesh 网的饱和请求率

VC 数量	1	2	3	4	5	6
饱和请求率	0.07	0.12	0.18	0.24	0.3	0.34

如果用 Chien 模型考虑虚通道对路由器建立延迟的影响, 令无虚通道 ($VC=1$) 时的建立延迟为 1 个时间单位, 模拟 4-ary 2-mesh 网的平均传输延迟的结果如表3。模拟结果显示: 虚通道数量并不是越多越好; 在某请求率条件下, 虚通道的数量将在某值处达到最优。

表3 4-ary 2-mesh 网的平均传输延迟

请求率	0.02	0.03	0.04	0.05	0.06	0.07
$VC=1$	14.5	16.2	19.1	23.0	33.6	97.6
$VC=2$	17.0	17.2	17.6	18.1	19.8	21.2
$VC=3$	17.5	17.5	17.5	17.6	17.9	18.1
$VC=4$	17.9	17.9	17.9	17.9	17.9	18.0
$VC=5$	18.2	18.2	18.3	18.3	18.2	18.3
$VC=6$	18.5	18.5	18.5	18.5	18.5	18.5

注: 有下划线的值为某请求率下的最小传输延迟

3 PAM 路由算法

考虑到路由器的建立延迟、算法的自适应性和无死锁性, 我们提出了 PAM 路由算法。为了与 PAM 算法进行比较, 首先介绍两种路由算法: 维序路由和平面自适应路由算法。

3.1 维序路由 DOR 算法

维序路由 DOR (Dimension Ordered Routing) 算法是一种确定性路由算法。其基本思想是: 给每维安排一个序号, 消息在网络中按维序递增或递减的顺序依次在各维中传输。

DOR 路由算法保证了消息在传输时无通道等待环的存在, 因此它是无死锁的。同时它不需增加虚通道, 硬件实现较简单。但使用 DOR 时, 消息只能沿固定路径传输, 网络容错能力不强。

对应路由器模型, DOR 路由器的建立延迟 T_{DOR} 为:

$$T_{\text{DOR}} = T_{\text{AD}} + T_{\text{ARB}} + T_{\text{CB}} \quad (2)$$

DOR 路由器交叉开关的大小为 3×3 , 因此 $P_{\text{DOR}} = 3; F_{\text{DOR}} = 3$ 。

3.2 平面自适应 PAR 路由算法

平面自适应 PAR (Planar Adaptive) 算法是一种部分自适应路由算法^[7], 其基本思想是: 将网络划分为 $n-1$ 个平面 $A_i (i \in [0, n-2])$, 每个平面由相邻两维 d_i 和 $d_{i+1} (i \in [0, n-2])$ 构成, A_i 和 A_{i+1} 平面共享 d_{i+1} 维。

PAR 路由算法中, 消息按单调顺序依次经过各个平面并且自适应地在各平面内传输。因此 PAR 是部分自适应路由算法。

平面自适应算法要求在 k -ary n -mesh 中, 每条物理通道上共享三条虚通道。对应路由器模型, PAR 路由器的建立延迟 T_{PAR} 为:

$$T_{PAR} = T_{AD} + T_{ARB} + T_{SEL} + T_{CB} + T_{VC} \quad (3)$$

PAR 路由器交叉开关的大小为 4×4 , 因此 $P_{PAR} = 4; F_{PAR} = 4; V = 3$ 。

PAR 路由器的交叉开关规模较小, 路由器建立延迟相对较小, 因此被认为是一种较好的部分自适应路由算法。

3.3 PAM 算法

考虑到算法的自适应性和路由器的建立延迟, 我们针对双工的 k -ary n -mesh 提出一种无虚通道, 无死锁的部分自适应路由算法 PAM (Partial Adaptive for Mesh), 其主要思想是:

1) 将 k -ary n -mesh 的每条通道用 $ch_{dim, dir}$ 表示, 其中 dim 表示该通道所在的维, 取值为 0 到 $n-1$; dir 表示通道的方向, 取值为 $+$, $-$ 。

2) 将网络划分为 sn 个子网 $SN_i (i \in [0, sn-1])$ 。

其中, $sn = \begin{cases} \lfloor (n-1)/3 \rfloor \times 2 + 1 & n \bmod 3 = 1 \\ \lfloor (n-1)/3 \rfloor \times 2 + 2 & n \bmod 3 = 2 \end{cases}$

若 $n=1$ $SN_0 = \{ch_{0,+}, ch_{0,-}\}$;

若 $n=2$ $SN_0 = \{ch_{0,+}, ch_{0,-}, ch_{1,+}\}$;

$SN_1 = \{ch_{1,-}\}$;

若 $n=3$ $SN_0 = \{ch_{0,+}, ch_{0,-}, ch_{1,+}\}$;

$SN_1 = \{ch_{1,-}, ch_{2,-}, ch_{2,+}\}$;

当 $n > 3$ 时,

若 $n \bmod 3 = 1$, $SN_0 = \{ch_{0,+}, ch_{0,-}\}$;

若 $n \bmod 3 = 2$, $SN_0 = \{ch_{0,+}, ch_{0,-}, ch_{1,+}\}$;

$SN_1 = \{ch_{1,-}\}$;

其余每三维构成两个子网。

同 PAR 算法相似, PAM 算法可分为高层算法和低层算法。高层算法: 子网间使用确定性算法, 消息从 SN_0 依次通过各个子网。低层算法: 消息在子网中完全按照自适应的最短路径算法进行传输。

PAM 算法的具体描述为:

输入: 消息源结点 $S = (s_{n-1}, \dots, s_1, s_0)$;

消息目的结点 $D = (d_{n-1}, \dots, d_1, d_0)$;

初始化: 路由标志 R

$R = (r_{n-1}, \dots, r_1, r_0) = (d_{n-1} - s_{n-1}, \dots, d_1 - s_1, d_0 - s_0)$;

函数 $nch(j)$: 求子网 SN_j 中三种通道的维号 i ;

即 $i = nch(j)$ if $((ch_{i,+} \in SN_j) \parallel (ch_{i,-} \in SN_j))$

1) 修改 R : 若消息来自 $ch_{i,+}$, 则 $r_i = r_i - 1$;

若消息来自 $ch_{i,-}$, 则 $r_i = r_i + 1$;

2) if $(R=0)$ then 消息到达目的结点并被接收;

3) 选取消息需要在其中传输的最小子网号。

即: $j = \min(\text{num_sn})$

$\text{num_sn} \in [0, sn-1]$, 且 $r_i \neq 0$ {其中 $i = nch(j)$ }

4) if $((SN_j \text{ 中通道 } ch_{i,(ch_{i,+} \text{ 可用})} \&\& (r_i > 0)))$

• 72 •

if $((ch_{i,-} \text{ 可用}) \&\& (r_i < 0)))$;

$(SN_i \text{ 其他通道 } ch_{dim,(ch_{dim,+} \text{ 可用})} \&\& (r_{dim} > 0))$

$\parallel ((ch_{dim,-} \text{ 可用}) \&\& (r_{dim} < 0)))$

then 向该通道发出请求 Req;

5) if (Req 被响应) 经过相应通道将消息传向相邻结点; else 返回 3)。

PAM 路由算法借鉴了 PAR 算法, 将三种通道组成一个子网。PAM 路由算法具有如下特点: 1) 无虚通道; 2) 每一个子网中均不存在环路。无通道等待环是路由算法无死锁的充分条件^[6], 因此 PAM 算法是无死锁的; 3) 消息在每个子网中按完全自适应路由算法传输, 在子网间按确定性路由算法传输, 因此 PAM 算法是部分自适应的。

由 $dim_{0,+}$, $dim_{1,-}$ 和 $dim_{0,+}$ 构成的路由器结构如图 2。对应路由器模型, PAM 路由器的建立延迟 T_{PAM} 为:

$$T_{PAM} = T_{AD} + T_{ARB} + T_{SEL} + T_{CB} \quad (4)$$

PAM 路由器交叉开关的大小为 4×4 , 因此 $P_{PAM} = 4; F_{PAM} = 4$ 。

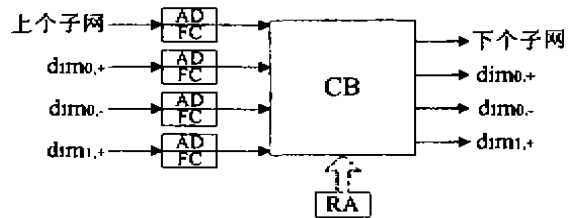


图2 PAM 路由器模型

4 性能分析

互联网的性能通常用消息传输延迟来衡量。消息传输延迟由两部分组成: 消息经过网络结点的时间和消息在结点处的等待时间。

通过前面几节的分析可以得出路由器建立延迟的理论结果, 但建立延迟只能反映消息经过结点的时间, 而不能反映消息在结点处的等待时间。因此我们对使用这三种路由算法的互联网进行了模型模拟。

4.1 理论结果

影响路由器性能的因素主要有: 输入输出端口数 P , 输入端路由时可选择的端口数 F 以及虚通道的个数 V 。将其参数代入公式 (2) 至 (4), 不同路由算法的建立延迟见表 4。

表4 建立延迟 $T(ns)$

T_{DOR}	T_{PAR}	T_{PAM}	延迟 = T_{DOR}/T_{PAM}	延迟 = T_{PAR}/T_{PAM}
5.6	10.9	8.7	0.64	1.25

从表中可以看出, PAM 的建立延迟小于 PAR 的

建立延迟,由于DOR不使用虚通道,且不需对路径进行选择,因此其建立延迟最小。

4.2 模拟结果

基于Chen路由器模型的参数,假定PAM路由算法下,消息经过互联网节点的延迟为1个时间单位(Delay=1),我们采用文[1]中的模拟器,分别对这三种路由算法进行了模拟,模拟中假定消息包含的消息微片个数为10。对16-ary 2-mesh的模拟结果见图3。图中X轴代表消息的平均请求率 P_{in} ,Y轴代表消息的平均传输延迟。

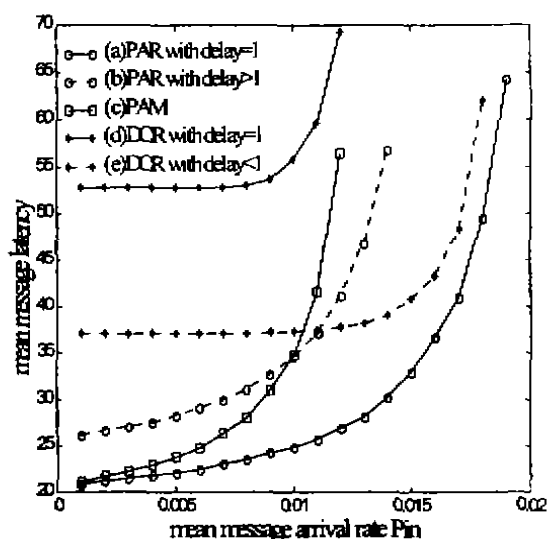


图3 16-ary 2-mesh的模拟结果

我们模拟的五种情况分别为:(a)结点延迟 Delay=1的PAR路由算法;(b)结点延迟 Delay>1的PAR路由算法;(c)PAM路由算法;(d)结点延迟 Delay=1的DOR路由算法;(e)结点延迟 Delay<1的DOR路由算法。从结果中可以看出,当建立延迟相同时(Delay=1),PAR路由算法的性能最优。这是由于PAR算法可以提供更多的路径来提高网络的吞吐率。由于PAM也提供了一定的可选择路径,所以PAM的性能优于DOR的性能。

当建立延迟不同时,从结果中可以看出:当 P_{in} 小于某一阈值(0.01)时,PAM路由算法的性能最优。此时对于PAR来说,建立延迟的影响大于虚通道提供的自适应能力带来的好处,使得它的性能比PAM的性能差。同时虽然DOR的建立延迟小,但消息不能根据网络负载选择路径,容易造成阻塞,使得它的性能也比PAM的差。

当 P_{in} 大于这一阈值时,PAM的性能已经不是最

优。此时,由于网络的消息请求率逐渐大于消息的服务率,网络逐渐进入不稳定状态。因此,当网络工作在稳定状态时,PAM算法是一种较好的路由算法。

结论 基于虫孔流控制的自适应路由算法一般采用虚通道来提高网络吞吐率和保证无死锁。但是,由于对应路由器交叉开关规模的增加,导致消息的建立延迟增加,从而使消息的网络延迟增加,降低了网络的性能。

本文基于k-ary n-mesh,综合了维序路由算法和平面自适应路由算法,提出了一种无虚通道、无死锁的部分自适应路由算法PAM。基于Chen模型,分别对维序路由DOR、平面自适应PAR和PAM路由算法进行了分析与模拟。

理论分析和模型模拟结果表明:

1)PAM的建立延迟大于DOR的建立延迟、小于PAR的建立延迟;

2)不考虑路由器延迟的影响时,PAR的性能最优,PAM的性能优于DOR的性能;

3)当消息请求率小于某一阈值时,PAM的性能最优。

PAM路由算法不需要虚通道即可以实现无死锁,且具有一定的自适应能力。综上所述,PAM是一种较好的路由算法。

参考文献

- 1 Xiao X Q(肖晓强), et al. Wormhole Routing Analysis of Bi-directional k-ary n-mesh. In: Proc. of 1999 Summer Computer Simulation Conf. 1999, to be published
- 2 Greenberg R I, oh HC. Universal Wormhole Routing. IEEE Trans. on Parallel and Distributed Systems, 1997, 8 (3): 254~262
- 3 董迎飞,等. 分布存储多机系统中的消息传递技术. 计算机研究与发展, 1994, 31(7): 1~9
- 4 Ni L M, McKinley P K. A survey of wormhole routing techniques in direct networks. Computer, 1993(2): 62~76
- 5 Dally W J. Virtual-channel flow control. IEEE Trans. on Parallel and Distributed Systems, 1992, 3(6): 194~205
- 6 Dally W J, Aoki H. Deadlock-free adaptive routing in multicomputer networks using virtual channels. IEEE Trans. on Parallel and Distributed Systems, 1993, 4(4): 466~475
- 7 Chien A A, Kim J H. Planar-adaptive routing: low-cost adaptive networks for multiprocessors. In: Proc. of the Intl. Symp. on Computer Architecture. 1992. 268~277
- 8 Duato J. A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing in Wormhole Networks. IEEE Trans. on Parallel and Distributed Systems, 1995, 6(10): 1055~1067
- 9 Glass C, Ni L M. The Turn Model for Adaptive Routing. In: 19th Annual Intl. Symp. on Computer Architecture. 1992. 278~287
- 10 Chien A A. A Cost and Speed Model for k-ary n-cube Wormhole Routers. In: Proc. of the Hot Interconnects Workshop. 1993