直接互连结构在数据交换中的应用分析

朱旭东* 李乐民

(电子科技大学 宽带光纤传输与通信系统技术国家重点实验室 成都 610054)

【摘要】研究了直接互连结构DIN应用到数据交换中存在的问题。针对数据交换应用特性，对DIN中的吞吐量、时延、路由算法和死锁解决策略进行了分析。研究表明，DIN中的理想吞吐量与结构规模的倒数成正比。把DIN应用到数据交换结构中，源路由表技术和最短路径算法更适合，采用死锁恢复机制能更好的利用系统资源。

关 键 词 直接互连结构；交换结构；路由算法；死锁

中图分类号 TN915.02 文献标识码 A

Analysis of Switching Fabrics Employing Direct Interconnection Networks

Zhu Xudong Li Lemin

(State Key Laboratory of Broadband Optical Fiber Transmission and Communication Networks, UEST of China Chengdu 610054)

Abstract In this paper, based on the characteristics of packet switching fabrics, performance of direct interconnection network have been analyzed in term of throughput, latency, routing algorithm and deadlock free. The research shows that ideal throughput is inversely proportional to the fabric scale. In the switch fabric employing direct interconnection networks, source routing algorithm and shortest path are more suitable in packet switching fabric. Furthermore, deadlock recovery mechanism has better utilization of resource.

Key words indirect interconnection network; switching fabric; routing algorithm; deadlock

直接互连结构(direct interconnection network，DIN)是处理节点通过一定方式互连构成的一个系统。直接互连结构中的处理节点是业务源，同时又是转发节点。图1是一个圆环(torus)直接互连结构。图中，圈表示处理节点，也称为交换节点或路由单元。DIN在大型多处理器系统(multiprocessors)、多个计算机间的互连和可扩展的共享缓存多处理器系统中有着广泛应用。其结构具有较高的并行处理(parallelism)特性和极好的扩展性。而数据交换应用则是指在LAN或WAN中使用的技术，例如ATM技术,TCP/IP技术等。本文把DIN技术应用到数据交换中，来实现高性能的数据交换结构。

表1列出两种应用主要性能指标。可看出数据交换中对吞吐量的要求更高，需要有业务质量保证，但对分组时延要求不高。数据交换结构需有无阻塞特性，输出到不同端口间的业务不会相互影响，应保证并发的多个业务同时以请求的带宽传送。

DIN具有较好的可扩展性和分布式特性，但不同端口间业务相互影响很大，造成结构吞吐量低。随着VLSI技术的发展，在硬件上可以实现更大的缓存和更高的时钟频率及数据传送速率，使DIN应用到数据交换成为可能。

文献[1]比较了传统数据交换结构与DIN在拓扑属性上的差别，给出了一种易扩展的数据交换结构。文

2002年12月20日收稿

* 男 29岁 博士生 主要从事可扩展的高速交换结构，交换结构中的路由算法和调度算法方面的研究

献[2]仿真分析了一种直接互连结构与几种典型数据交换结构的性能比较，在文献[3]中实现了一种应用到数据交换中的路由器结构，分析了实现路由器的折中方案。

本文旨在基于DIN结构的吞吐量、时延、路由算法和死锁解决进行了分析。得出了一些在应用直接互连交换结构构建数据交换结构时可以采用的方法。

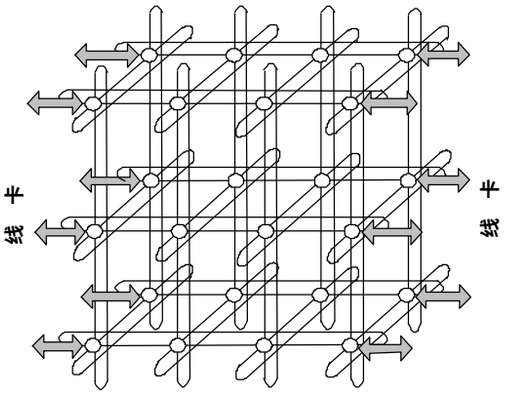

图1　4×3×2 torus交换结构图

表1　并行处理与数据交换应用特性比较

参数	并行处理	数据交换
端口数	1～2 048	4～512
峰值速率	4 Gb/s	10 Gb/s
平均速率	100 Mb/s	7 Gb/s
数据单元长度	64或512bit	40～1.5 kB
时延要求	100 ns	10 μs
时延抖动	不要求	要求
包丢失率	不允许	$<10^{-12}$
业务服务质量		需要
拥塞控制	自控制	非自控制

1　交换结构吞吐量和时延

DIN中两个重要的特性是吞吐量和时延。吞吐量是指每个输入端口接受的最大数据速率(通常是bit/s)。时延是指分组从源到目标节点所需总时间。吞吐量和时延不仅同拓扑结构有关，还与采用的路由算法、业务源模型和流量控制策略等有关，本文主要分析理想吞吐量和无冲突时延。理想吞吐量指路由算法能完全平衡从源到目标所有路径上的负载，对流量控制能保证瓶颈通道上不存在空闲周期。无冲突时延指无内部阻塞情形下分组所需传输时间。

通道c的负载强度l_c是指每个节点以随机方式选择一个目标节点发送一个分组，经过通道c的分组平均个数。或是每个节点向所有节点(包括自己)发送分组时经过通道c的分组个数除以交换节点数。实际可以采用下面的等式来计算通道负载强度

$$r_c = \frac{l_c}{l_{in}} \tag{1}$$

式中　l_{in}为输入单个节点的业务量，l_c为经过通道c的业务量。最大负载强度为

$$r_{\max} = \max(r_c) \tag{2}$$

拓扑结构的最大通道负载强度$r_{\max}$同通道物理带宽b结合，可以用来确定理想吞吐量的值。整个结构的理想吞吐量为

$$q_{ideal} = \frac{b}{r_{\max}} \tag{3}$$

式中　b是物理链路的带宽，式(3)是在均匀分布输入业务的计算公式。

计算双向通道Torus结构的理想吞吐量。设输入负载r_{in}，i维上节点个数k_i，则i维上的平均跳数就$\overline{k_i} = k_i / 4$。i维上总负载容量为$l_i = \overline{k_i} k_i l_{in}$。由于总通道数是$2k_i$，所以每个通道的负载是$l_c = \overline{k_i} k_i l_{in} / 2k_i$。代入式(1)得到：$r_c = k_i/8$。结合式(2)、(3)得到理想吞吐量$q_{ideal} = 8b / k_i$。由此公式发现随着规模$k_i$增加，如物理通道带宽$b$保持不变，理想吞吐量下降。这是针对无虚拟通道[4]、无自适应性给出的理想吞吐量。为了保证随着结构规模增加，吞吐量不变或下降较慢，需要采用理想的路由算法来实现。

交换结构的无冲突时延包括三部分：节点处理时延，通道传播时延和传送时延。H_{avg}表示节点个数，t_r是指每个节点处理时间，D_{avg}表示平均传播通道长度，v是指传播速度，L表示分组长度

$$T_{avg} = H_{avg} t_r + \frac{D_{avg}}{v} + \frac{L}{b} \tag{4}$$

2 路由算法

路由算法用来在特定的拓扑中找到一条或多条从源至目标节点的路径。维序算法(dimension order routing)以确定的维顺序路由，在Mesh结构中不需要虚拟通道就可以实现无死锁路由[4]。它实现简单，但性能差，特别在业务负载不均匀，交换结构规模较大时更为突出。自适应路由算法通过寻找空闲通道的方法来平衡负载，增加整个结构吞吐量。文献[5]提出了无死锁的部分自适应的最短路径路由算法(planar adaptive algorithm，PAR)：对任意规模的Mesh结构都只需3个虚拟通道就可解决死锁，该算法具有一定的自适应特性。如果该算法应用在Torus中，由于Torus结构较Mesh结构在每维最远节点对间存在环路，所以还需提供一倍的虚拟通道来防止死锁。

文献[6]提供了一种自适应路由算法的构造方法。是对阻塞的分组提供额外逃逸通道。在分组出现阻塞后采用该通道，同时在逃逸通道所构成的网络中采用一定资源限制防止死锁。比PAR进一步增加了自适应特性。

Star_channel算法最多用5个虚拟通道来防止死锁[7]。具有完全自适应性的最短路径算法，即路由算法可应用从源节点s到目标节点d的所有最短路径。

本文提到的三种算法都是动态自适应的，根据网络的状态来改变路由。还有一种是与网络状态无关的路由算法，如下所述。

文献[8]算法通过随机选择象限和中间节点的方法来平衡负载。RLB算法在此作了进一步改进[9]，能够保证在同一维上的负载平衡。由于与结构状态无关，此算法可采用源路由表方法来实现。在数据交换结构中源路由表技术更适合。路由算法简化了中间节点及更高处理速度的实现，提高了整个结构吞吐量。而且使用源路由表时，改变路由设置简单，只需更新源节点上的路由表。

本文所述的算法都是通过对资源的限制来防止死锁，充分利用物理链路带宽，平衡各个物理链路上的负载，但是对于节点上的缓存资源利用率不高。采用死锁恢复的策略可以解决这个问题。

3 死锁避免和死锁恢复

由于存在占用资源者申请另一个资源的情形，在DIN中由于拓扑结构本身存在环状路径，所以在DIN中易死锁。图2所示，A,B,C,D是4个分组的路径(Path)构成循环形成死锁的情形。文献[4]讨论了互连结构中死锁的问题，提出了用通道依赖图(channel dependency graph，CDG)是否存在循环来判断是否死锁。文献[6]改进了这种方法，肯定了CDG图的充分性和必要性，即动态出现CDG图循环时不死锁。

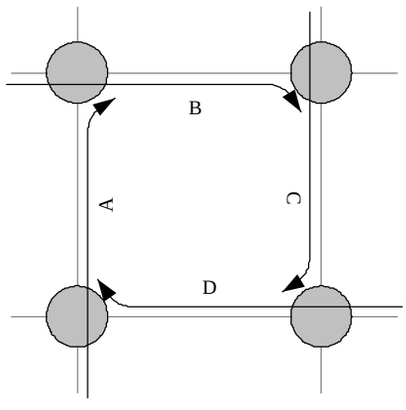

图2 A,B,C,D 4个分组构成一个死锁

解决死锁的方法可分为两大类：死锁避免和死锁恢复。死锁避免是指通过在路由算法设置禁用某些资源来防止出现路径环。文献[10]通过限制一个方向上的路由来打破循环，避免死锁。本文2中路由算法也都采用死锁避免策略。死锁避免是通过限制对一些资源(缓存或物理通道)的利用来防止死锁，因此交换结构资源利用率不高。

死锁恢复是在发现结构内发生死锁后，通过一定的策略来解决死锁。允许分组使用交换结构中的任何资源，提高交换结构的资源利用率。一般有两种死锁恢复方法。消极死锁恢复法，采用丢弃，偏移路由来处理死锁分组。积极死锁恢复法通过分配其他资源来传送死锁分组到达目标节点。对于死锁的判决是死锁恢复机制中的一个难点，特别是在业务负载和分组长度变化范围大时。如何快速区别分组是阻塞还是死锁。在文献[11]中讨论了基于等待时间的死锁恢复机制。文献[12]介绍了一种分布式死锁检测机制，它采用判断阻塞分组中根节点的方法来降低误判率。

在数据交换结构中采用死锁恢复机制更适合在数据交换技术中，整个结构资源有限，数据交换对于传送带宽的要求更高。数据恢复机制需要结合自适应路由算法来降低死锁的发生率。因此分析各种路由算法的死锁发生率是十分必要的。

4 结 束 语

本文对数据通信交换结构中的互连结构应用进行了讨论，论述了在并行处理技术中的路由算法，提出在数据交换结构应用中对路由算法的要求。随着结构规模的扩展，整个结构的吞吐量将下降，其理想吞吐量的值与结构规模的倒数成正比，故源路由表技术和最短路径算法更适合在数据交换结构中应用。另外，分析了数据交换中的死锁解决策略，若有好的死锁检测机制，则死锁恢复机制更适合在数据交换结构中应用。

参 考 文 献

[1] Dally W J. Scalable switching fabrics for internet routers[R]. Computer Systems Laboratory, Stanford University and Avici Systems Inc, 1999

[2] Nader F M, An efficient switching fabric for next-generation large-scale computer networking[J]. Elesevier computer Networks, 2002,40(2): 305-315

[3] Duato J, Yalamanchillli, Caminerl, *et al.* MMR: a high performance multimedia router – architecture and design trade-offs[C]. Proc 5th symp. On High Performance Computer Architecture(HPCA-5), 1999. 300-309

[4] Dally W J. Virtual channel flow control[J]. IEEE Trans. Parallel and Distributed Systems, 1992, 3(3): 194-205

[5] Chien A A, Kim J H. Planar adaptive routing: low-cost adaptive networks for multiprocessors[J]. J. ACM, 1995, 42(1): 91-123

[6] Duato J. A new theory of deadlock-free adaptive routing in wormhole network[J]. IEEE Trans. Parallel and Distributed Systems, 1993, 4(12): 1 320-1 331

[7] Gravano L, Gusatavo D. Adaptive deadlock- and livelock-free routing with all minimal paths in torus networks[J]. IEEE Trans. Parallel and Distributed systems, 1994, 5(12)

[8] Nesson T, Lennart S, Johnsson. ROMM routing on mesh and torus networks[C]. In Proc. 7th Annual ACM Symposium on Parallel Algorithms and Architectures SPAA' 95, Santa Barbara, California, 1995. 275-287

[9] Singh A, Dally W J. Locality-preserving randomized oblivious routing on torus networks[R]. Stanford University, 2002

[10] Glass C J, Ni L M. The turn model for adaptive routing[J], J. ACM, 1994, 41(5): 874-902

[11] Petrini F, Vanneschi M. Performance analysis of minimal adaptive wormhole routing with time dependent deadlock recovery[C]. International Parallel Processing Symposium Conference Proceedings 1997

[12] Pinkston T M. Flexible and efficient routing based on progressive deadlock recovery[J]. IEEE Transactions on Computers, 1999, 48(7)

编　辑　孙晓丹