两级分布式共享存储器结构及算法*)

伊 鹏 罗敏轩 靳毓国 郭云飞

(信息工程大学 国家数字交换系统工程技术研究中心 郑州450002)

摘 要 商用存储器的随机访问速率和交换结构的交换速率一直是限制高速路由器性能提高的主要因素,改善交换结构使其降低对存储器存取速率的要求是解决问题的关键。本文提出了两级分布式共享存储器(TSDSM)结构,给出了该结构模拟输出排队调度算法所需存储器个数的下界及相应的模拟FCFS与PIFO输出排队算法,并对算法模拟输出排队算法的可行性给出了证明。TSDSM结构及相应的算法控制机制不仅使目前商用存储器的存取速率能满足交换速率的要求,而且交换结构可以工作于非加速状态。

关键词 交换,调度,输出排队,DSM

A Two-Stage Distributed Shared Memory Architecture and its Scheduling Algorithms

YI Peng LUO Min-Xuan JIN Yu-Guo Guo Yun-Fei

(NDSC, Information Engineering University, Zhengzhou 450002)

Abstract The capacity of high speed packet switch is limited by the random access time of commercially available memories and the rate of switch architecture. The key point to solve such a problem is improving switch architecture and using slow memories to realize high performance packet switch. In this paper, we propose a two-stage distributed shared memory architecture (TSDSM). The lower bound of it is also given. Scheduling algorithms for a TSDSM imitating a FCFS output-queued (OQ) switch and a PIFO OQ switch are given too. The validities of these algorithms are theoretically proved. Without speedup the TSDSM can be used to realize high speed packet switch with commercially available memories.

Keywords Switch, Scheduling, Output queue, Distributed shared memory

1 引言

波分复用(WDM)技术的发展及应用极大地提高了计算机网络的传输能力,这对于路由器的性能提出了更高的要求,然而高速路由器发展一直受商用存储器的随机访问速率和交换结构交换速率的限制。目前商用存储器的随机存取速率是难以满足传输技术发展要求的,当高性能分组交换机的端口速率大于OC192(10Gb/s)达到OC768(40Gb/s)甚至OC3072(160Gb/s)时,当前使用的交换机几乎不可能实现对分组包的缓存,严重影响了路由器性能的进一步提高。对此,本文研究了新的交换结构,以降低对存储器访问速率的要求,从而使得路由器的速度及性能得到提高,以满足网络的传输要求。

前人在这方面也做了一定的工作,Sundar Iyer 和 Nick McKeown 等人在文[1,2]中提出了并行分组交换(PPS—parallel packet switch)结构,该结构可以降低对存储器访问速率的要求,但达到该目的是以交换结构的增多为代价的,并且多个交换结构的同时使用,使得该系统的实现复杂度及费用都有较大的增加,同时交换机的整机性能也难以维护。在2002年的研究中,Sundar Iyer 又提出了一种分布式共享存储器(DSM—Distributed Shared Memory)结构[3],该交换结构的本质是在交换机的各输入端口中,根据业务分配理论[4~6],将到达数据包业务负荷分担到一个共享的存储器组中来模拟输出排队[7]。该交换结构的实现不但简单经济,并且能够降低交换机对存储器存取速率的要求。然而DSM结构仍然需要其中的Crossbar以四倍甚至八倍的加速比[3]工作于加速状态,这在实际的设计中是难以实现的。

为此我们提出了两级分布式共享存储器(TSDSM—Two Stage Distributed Shared Memory)结构,该结构是在DSM结构的基础上增加了一级Crossbar。相对于存储器而言,TSDSM结构在其输入输出端口分别使用了一个Crossbar。与DSM结构相比,TSDSM结构的优点在于可负荷分担交换机的到达业务,从而降低其对存储器存取速率的要求。而且该结构在存储器的输入输出端口上分别实现Crossbar的分离设计,使得Crossbar可以不工作于加速状态。此外,基于此结构我们提出了相应的模拟FCFS(first come first serve)及PIFO(push in first out)输出排队调度算法,并且从理论上证明了在该结构下这几种算法的可行性与有效性,给出了相应的结论。值得指出的是,文[8]中所给出的输出排队调度算法中所采用的交换结构虽然与TSDSM相似,但是其中只讨论了端口速率与存储器的存储速率相同时的情况,仅相当于本文所讨论的结构模型中的一种特例。

2 TSDSM结构模型

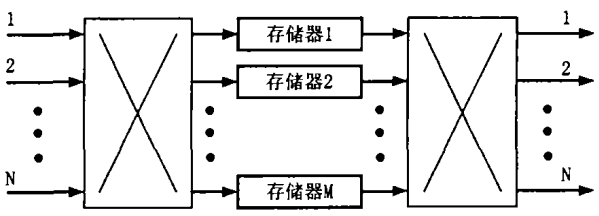

图1 TSDSM结构模型

*)国家863课题基金(No. 2001-AA-12-4-011)。**伊 鹏** 博士生,主要研究方向为路由器交换调度技术。**罗敏轩** 工程师,主要从事路由交换技术的研究。**靳毓国** 学士,现为信息工程大学电教中心教员,主要从事网络技术研究。**郭云飞** 教授,博士生导师,863计划信息技术领域通信技术主题专家组组长。

TSDSM 结构模型如图1所示，由 IP 分组包切片后形成的等长数据包，其长度一般为512bit 或64Byte，我们称之为 Cell，经过 N×M 的 Crossbar，写入 M 个读写速率均为 r 的存储器进行排队后，通过 M×N 的 Crossbar 输出，输入及输出端口的线路速率均为 R。在线速率为 R 的条件下，将发送或接收一个 Cell 的时间称为系统时隙，存储器读入或写出一个 Cell 的时间称为存储器时隙，而读写速率均为 r 的存储器读入或写出一个 Cell 所需的系统时隙数称之为存储周期，在数值上，存储周期等于$\lceil R/r \rceil$。在 DSM 结构中，存储器的输入与输出都要经过同一个 Crossbar，这样使得该 Crossbar 上的交换任务过于繁重，同时原有算法的效率较低，使得其工作于4至8倍的加速状态。此外由于该 Crossbar 要同时完成输入与排队输出双重业务，控制算法在实现上也相当复杂。由图1中所给出的结构模型可见，TSDSM 结构在存储器的输入端口与输出端口各使用了一个 Crossbar，这样对存储器的读、写就可独立地控制，并且这两个 Crossbar 可以分担 cell 的交换任务。如果在 Crossbar 的控制算法上采取有效的控制机制，可以实现在降低系统中存储器访问速率的同时，使 Crossbar 不用工作于加速状态。文[8]中所述输出排队调度算法采用的交换结构虽然与 TSDSM 相似，但其结构仅仅是本文的 TSDSM 结构中 R=r 时的一种特例。在本文所分析的结构下，存储器的读写速率 r 在理论上可以降到任意低。

就 TSDSM 结构实现模拟输出排队调度算法的一般情况，为了能够有效降低对存储器访问速率的要求，输入端口的到达 cell 会被负荷分担到一个速率较低的分布式存储器组，当 cell 的离去时间到达时，cell 会从存储器组读出并送往输出端口。因此在 TSDSM 结构上模拟一般的输出排队调度算法时，每一个到达 cell 将面临以下四种冲突：1. 在 cell 写入的一个存储周期内存储器不能被再次写入；2. 同一系统时隙到达的 cell 不能被送往相同的存储器；3. 在读取 cell 的一个存储周期内存储器不能被再次读取 cell；4. 同一系统时隙不能从相同的存储器读取 cell。

为了使 TSDSM 结构能够模拟输出排队调度算法，一方面 TSDSM 结构中的存储器在数量上必须满足一定要求，另一方面需要引入一定的算法控制机制，才能够使得 TSDSM 结构避免发生以上所分析的四种冲突中的任何一种。通过对 TSDSM 结构分析，对于 TSDSM 结构中的存储器数量，在定理1中给出了所需满足条件的下界。

定理1 要使 TSDSM 结构能模拟输出排队调度算法，其结构中存储器数量至少应不小于$2\lceil R/r \rceil N-1$。

证明：考虑 TSDSM 结构中任意一个到达 cell 的情况。在任意一个系统时隙，冲突1最多将导致分布式存储器组中的$(\lceil R/r \rceil-1)N$ 个存储器无法写入 cell；冲突2最多致使$(N-1)$个存储器无法写入 cell。对于 cell 的离去过程有类似的考虑：在任意一个系统时隙，冲突3最多将导致分布式存储器组中的$(\lceil R/r \rceil-1)N$ 个存储器无法读取 cell；冲突4最多致使$(N-1)$个存储器无法读取 cell。最坏情况对应于冲突1，冲突2，冲突3及冲突4中无法读写的存储器数目均取最大值且相互之间的交集均为空集，此时只有当存储器的数量不小于$2\lceil R/r \rceil N-1$时才可能存在可避免冲突的控制算法。因此要使 TSDSM 结构能模拟输出排队调度算法，其存储器的数量至少应不小于$2\lceil R/r \rceil N-1$。证毕。

定理1得出了 TSDSM 结构模拟输出排队调度算法时，分布式存储器组所需存储器个数的下界，但要在 TSDSM 结构上模拟输出排队的调度算法，还必须对其控制机制进行进一步的优化，使得通过算法控制能避免冲突的发生。为此我们基于 TSDSM 结构，提出了相应的模拟 FCFS 输出排队调度算法与模拟 PIFO 输出排队调度算法，并从理论上证明了该算法的有效性与可行性，从而使得该结构能充分模拟采取不同机制的输出排队调度算法。

3 模拟 FCFS 输出排队调度算法

为了方便分析比较，我们引入一个 N×N 的参考交换结构，其输入输出端口速率均为 R，采取输出排队方式，其中每一输出队列可缓存 L 个 cell。我们首先考虑模拟采取 FCFS 机制的输出排队调度算法。对于采用 FCFS 机制的参考交换结构，假定到达 cell 按如下规则进行服务：对于所有输出队列中到达时间不同的 cell，根据 cell 到达时间的先后进行服务；对于任一输出队列中到达时间相同的 cell，以到达 cell 的输入端口序号为序进行服务。对于 TSDSM 结构模拟采取 FCFS 机制的输出排队调度算法，在假定存储器数量满足定理1要求的前提下，为避免冲突的发生，还需引入必要的控制机制。为此我们提出如下算法：

算法1

1. 根据参考交换结构中 FCFS 的服务规则确定到达 cell 的离去时间，并由 cell 离去时间确定冲突3和冲突4的冲突域。

2. 根据冲突1、冲突3和冲突4的冲突域确定到达 cell 的可用存储器组，即在一个存储周期内没有发生 cell 读入和写出事件，且不含有与到达 cell 相同离去时间的一组存储器。

3. 在到达 cell 与可用存储器组之间确定其匹配子图，并依据此匹配关系将到达 cell 写入存储器组。

4. 存储器组中的 cell 的离去时间到达时即通过第二级 crossbar 送往输出端口。

参照文[7]，在 TSDSM 结构中，如果对于在任一系统时隙到达的任一 cell，都可以按与采取 FCFS 机制的参考交换结构相同 cell 离去顺序发送出去，则可认为该结构可以模拟采取 FCFS 机制的输出排队调度算法。对算法1进行分析，得到如下定理：

定理2 满足定理1的 TSDSM 结构在算法1的控制下可以模拟采取 FCFS 机制的输出排队调度算法。

证明：根据我们所采用的算法控制机制可知：对于满足定理1的 TSDSM 结构，如果任一到达 cell 在算法1的控制下不会被阻塞于系统中，则可以按与采取 FCFS 机制的参考交换结构相同的 cell 离去顺序发送出去。算法1第2步所得的可用存储器组排除了发生冲突1、冲突3和冲突4的可能性，第3步根据可用存储器组计算匹配子图则进一步排除了发生冲突2的可能性。因此到达 cell 不可能被阻塞于系统中。所以满足定理1的 TSDSM 结构在算法1的控制下可以模拟采取 FCFS 机制的输出排队调度算法。证毕。

FCFS 机制只是基于输出排队结构调度算法的一种特殊情况，还有许多不同的基于输出排队结构的调度算法，如 WFQ[9]、VC[10]等，它们都属于采取 PIFO 机制的调度算法。PIFO 机制指的是新到达 cell 可以从队列的任一位置插入队列中，但只能从队列首部出队。TSDSM 结构模拟输出排队调度算法的关键是要能够通过控制，模拟采取 PIFO 机制的输出排队调度算法。

4 模拟 PIFO 输出排队调度算法

在 TSDSM 结构上模拟采取 PIFO 机制的输出排队调度算法时，前述分析的四项冲突依然存在。但是由于模拟的输出排队结构所采取的调度策略不同，使得避免冲突4的方法变得复杂。冲突4要求同一系统时隙不能从相同的存储器读取

cell。对于模拟采取 FCFS 机制的输出排队调度算法，由于到达 cell 的离去时间确定后不会改变，因此只需使相同离去时间的 cell 不写入到同一个存储器即可消除冲突。但是对于模拟采取 PIFO 机制的输出排队调度算法，由于 cell 可从队列的任意位置入队，因此队列中缓存 cell 的离去时间可变。虽然对于冲突4依然最多只有 N－1个存储器无法读取 cell，但我们无法确定这些存储器的位置，因而采用简单的机制无法避免冲突4。

对于冲突4，虽然理论上一个系统时隙中参考交换结构最多有 N 个存储器在读取 cell，但由于它采用的是 PIFO 机制，cell 到达时刻及其离去时间无法预知，使得 TSDSM 结构中潜在的冲突无法避免。考虑在往存储器写入某一 cell 记作 a 的情况，写入时我们只能根据当前系统时隙的情形，挑选不会导致冲突4发生的存储器，但由于 cell 离去时间的可变性，存储器中原有缓存 cell 的离去时间可能变得与 a 的离去时间相同，因而冲突4仍可能发生。用反证法考虑极端的情况可以证明，这种无法预知的冲突是无法通过这种方式完全避免的。因此我们考虑引入一定的乱序来避免冲突4的发生，由于引入的乱序是有限的，在输出端通过缓存采用简单的算法即可恢复原来顺序。基于这一考虑，我们在 TSDSM 结构的每一输出端口加一缓存，并提出如下两种算法。

算法2

1. 利用算法1将到达 cell 送往输出端口缓存。

2. 在输出端口缓存采取简单的排序算法进行乱序调整。

因为对于队列长度为 L 的参考交换结构，用 FCFS 的方式处理 PIFO 机制的业务，cell 最多被延迟 L 个系统时隙，也就是说其乱序不会超过 L。因此在输出端口可以通过乱序调整来实现 cell 的重新排序，使其按照参考交换结构中调度算法控制的 cell 离去顺序离开。

算法2是在算法1的基础上经简单改进得到的。由于算法2已引入了乱序，因此第一步完全按照算法1的方式处理其实是不必要的，而且使得算法自身过于复杂。在引入乱序调整的机制下，我们可以采用更简单的算法对 cell 进行调度，为此我们提出了简化的算法。

算法3

1. 根据冲突1和冲突2确定可写存储器组。

2. 按照连续 N 个相同目的端口的 cell 被写入不同存储器的原则，将到达 cell 写入到可写存储器组。

3. 根据冲突3确定可读存储器组。

4. 以 N 个系统时隙作为一个处理单元，在其中每一系统时隙确定可读存储器组和目的端口之间的匹配子图，cell 按匹配关系被读出到输出端口缓存。

5. 在输出端口缓存采取简单的排序算法进行乱序调整。

对于队列长度为 L 的参考交换结构，算法3中导致的乱序最大为2N－1，因此可通过乱序调整来实现 cell 的重新排序，使其按照参口交换结构中所采用具体调度算法控制的 cell 离去顺序离开。需要说明的是：算法3所需的存储器数目和前面相同。虽然算法3没有直接考虑冲突4，而是采取引入乱序的方式避免冲突4的发生，但是这一方式需要在写入过程中连续 N 个相同目的端口的 cell 不能被写入相同存储器，这样最多有 N－1个存储器无法写入 cell，在数量上刚好与直接避免冲突4一致，因此所需的存储器总数不变。

同前所述，如果在 TSDSM 结构中，如果对于在任一系统时隙到达的任一 cell，都可以按与采取 PIFO 的参考交换结构相同 cell 离去顺序发送出去，则我们认为该结构可以模拟 PIFO 的输出排队[7]。通过分析算法2和算法3，我们得到如下定理：

定理3 满足定理1的 TSDSM 结构在算法2或算法3的控制下可以模拟采取 PIFO 机制的输出排队调度算法。

证明：根据我们所采用的算法控制机制可知：对于满足定理1的 TSDSM 结构，如果任一到达 cell 在算法2或算法3的控制下，不会被阻塞于系统中，则可以通过乱序重排按与采取 PIFO 机制的参考交换结构相同 cell 离去顺序发送出去。因为算法2和算法3引入的乱序都是有限的，因此 cell 不会被阻塞于乱序重排的缓存中。对于算法2，由定理2的证明可知：在其控制下，到达 cell 不会被阻塞于系统中。对于算法3，根据其第一步所得的可写存储器组排除了发生冲突1和冲突2的可能性，根据第三步所得的可读存储器组排除了发生冲突3的可能性。算法第二步采用引入乱序的方式避免了冲突4的发生，因此到达 cell 也不会被阻塞于系统中。所以满足定理1的 TSDSM 结构在算法2或者算法3的控制下可以模拟采取 PIFO 机制的输出排队调度算法。证毕。

至此，我们已经证明了 TSDSM 结构模拟输出排队调度算法的一般情况。值得指出的是：在模拟采取 PIFO 机制的输出排队调度算法时，DSM 结构和 TSDSM 结构均导致了 cell 的乱序，虽然可通过采取乱序重排机制调整，但均会导致算法的时延性能有所下降。

结束语 商用存储器的随机访问速率和交换结构的交换速率一直是高速路由器实现的瓶颈，为此需要改善交换结构来降低对存储器存取速率的要求。虽然 DSM 结构能够较好地解决该问题，但其交换结构仍需以4倍乃至8倍的加速比工作。为此本文对交换结构进行了进一步研究，提出了 TSDSM 结构。为了能使该结构充分模拟输出排队调度算法，我们给出了其中并列存储器组所需的存储器个数的下界，同时给出了基于该结构的模拟 FCFS 输出排队和模拟 PIFO 输出排队的算法，并证明了 TSDSM 结构下这两种输出排队算法的可行性。研究结果表明 TSDSM 结构及相应的算法控制机制不仅能较好地解决了存储器的随机访问速率问题，使目前的商用存储器的存取速率满足交换速率的要求，同时在存储器的输入输出端口分离使用的 Crossbar 使得算法控制比较简单，而且可以使交换结构工作于非加速状态，为高速路由器的实现提供了一个较好的选择方案。

参 考 文 献

1 Iyer S, Awadallah A, McKeown N. Analysis of a packet switch with the memories running slower than the line rate [A]. In: Proc. of INFOCOM'2000 [C], Tel-Aviv, Israel, vol. 2: 529～537

2 Khotimsky D A, Krishnan S. Stability Analysis of a Parallel Packet Switch with Bufferless Input Demultiplexors [A]. In: Proc. of ICC 2001 [C], Helsinki, Finland, Jun. 2001

3 Iyer S, Zhang R, McKeown N. Routers with a Single Stage of Buffering [A]. SIGCOMM'02 [C], Pittsburgh, Pennsylvania, USA, 2002

4 Adiseshu H, Parulkar G, Varghese G. A reliable and scalable striping protocol [A]. In: Proc. ACM Sigcomm [C], 1996

5 Chiussi F, Khotimsky D, Krishnan S. Generalized inverse multiplexing of switched ATM connections [A]. In: Proc. IEEE Globecom '98Conference. The Bridge to Global Integration [C], Sydney, Australia, Nov. 1998

6 Chiussi F, Khotimsky D, Krishnan S. Advanced Frame Recovery in Switched Connection Inverse Multiplexing for ATM [A]. In: Proc. ICATM '99 Conference [C], Colmar, France, Jun. 1999

7 Iyer S. The Parallel Packet Switch Architecture [D]. M. S. Thesis Report, Stanford University, Stanford, USA, 2000

8 Prakash A, Sharif S, Aziz A. An O(log²N) parallel algorithm for output queuing [A]. IEEE INFOCOM 2002 [C], New York, USA, June 2002

9 Parekh A. A generalized processor sharing approach to flow control in integrated services networks: [PhD diss]. MIT, MA, 1992

10 Zhang L- X. Virtual clock: A new traffic control algorithm for packet switching networks. In: Proc ACM SIGCOMM' 90 , Philadelphia PA, 1990 . 19～29