国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第19卷第6期 1997年12月 Vol19 No 6

影响MPP通讯性能的因素分析及解决途径探讨*

肖立权 张民选 周兴铭

（国防科技大学计算机科学系 长沙 410073）

摘 要 本文分析了MPP中产生通讯延迟的几个重要因素，概述了这些方面的研究现状，给出提高通讯性能的一些策略和研究要点。

关键词 通讯，线程，互连网络，路由器

分类号 TP393

Analysis and Research on the Communication for Massively Parallel Systems

Xiao Liquan Zhang Minxuan Zhou Xingming

(Department of Computer Science, NUDT, Changsha, 410073)

Abstract This paper analyzes several important factors contributing to the communication latency in massively parallec processors, summarizes the research background and proposes some schemes to improve the communication performance.

Key words communication, thread, interconnection network, router

分布主存的MPP机相对于传统的向量机、共享主存的多处理机而言，提供了更高带宽的访存能力、更强的可伸缩性、更快速度的运算潜能。在MPP机上，大规模计算问题被分解成许多可以并发执行的单元（进程或线程），一个处理机上驻有一个或几个这样的执行单元，运行过程中，由于它们之间的依赖关系，需进行同步和通讯。同步通讯常用消息传递来实现，它是影响MPP机性能的关键。

影响通信性能的因素多种多样，LogP计算模型为算法设计和分析者提供了一个影响系统性能因素的简单机器抽象。本文从系统结构设计的角度，分析了产生通信延迟的几个主要因素，概述了这些方面的研究现状，并从消减OS的影响、增强互连网络的可伸缩

* 1996年3月14日收稿

性、设计高性能路由器等方面提出了提高通讯性能的一些策略和研究要点。

1 消减 OS 的影响

通信过程中的软件开销占据了延迟的很大一部分，操作系统对这部分延迟有着极大的影响，它与程序中的并行表达与控制密切相关。即使对一个粗粒度并行的程序，如果创建和管理并行的开销很大，会导致较低的并行效率，而对一个细粒度并行的程序，在并行创建和管理开销很小时，也会呈现出高的并发效率。

构造并行程序的方法之一是采用一组共享主存的传统 UNIX 进程，每个进程有单个地址空间和此空间上的执行流，但是这样的进程是为单CPU 环境多道编程方式下的系统而设计的，它们简单但低效，只能进行大粒度的并行。为进行通用的并行编程，传统的进程被线程所取代，线程将顺序程序流从传统进程的地址空间、I/O 描述符等分离出来，一个多线程进程在单个进程中具有两个以上的线程以及程序计数器，它们共享同一地址空间、分配的资源等，线程的创建与管理开销要比进程的小得多。线程可以在用户级或核心级支持，但二者都不尽人意。

用户级线程模型是在用户空间进行线程的生成，消除前后关系转换等线程管理的模型，线程管理的子程序通过一个连接到应用程序的 Run-time 库来实现。线程管理不需访问核级，能取得较高的性能，且实现灵活，不需核级修改。运行在传统进程环境上的用户线程，它将每个进程看成一虚拟处理机，并将之看成是自己控制下的物理处理机，实际上，物理处理机被多个虚拟处理机共享，因此诸如多道程序、I/O、页故障等因素，都会打破虚拟机与物理机的平衡性，会导致用户级线程的低效，甚至影响到正确性。死锁问题的解决在用户级线程的设计中也十分困难。

多处理机 OS（例如 Mach，TopaZ，V）在核级直接支持同一地址空间上的线程，采用核级线程来管理，保证了正确性，但是核级线程的管理原语开销尽管比传统 UNIX 进程管理开销低一个数量级，但是比用户级线程管理开销要高出一数量级[1]，同时核级线程的管理也缺乏灵活性。因此采用用户级线程是获取高性能的最终手段，但这些用户级线程应在核级线程之上实现，而不是建立在传统有 UNIX 进程上，同时用户空间应能看到核空间的活动，这些要求导致了协调型用户级线程模型研究的必要性，Washington 大学提出称为 Scheduler activation 的机制，Rochester 大学提出与上述功能相似的线程模型。

MPP 上的 OS 为应用程序掩盖了硬件属性，它具有创建、消除、管理运行单元、分配存储单元、保证访问不越界、处理处理机内部以及处理机之间的通讯、组织 I/O 等功能。并行 OS 正从 hosted system 、symmetric system 向微核系统转化。微核系统使得每个节点不再是要么有一个大 OS，要么并无 OS，而是有一部分核心代码，提供基本的进程管理、存储器管理、通讯和 I/O 服务，而其它服务（如文件系统）由运行在用户态的 Server 进程提供，这种工作的划分，使得微核小而快。然而微核的功能划分，其构造方式仍是值得研究的重要问题，它直接影响着并行效率。MANNA 上的 PEACE 操作系统是微核构造方式探索的很好范例[2]。为减少通信中的软件开销，本文认为有几个方面值得深入探讨：

（1）并行的表达和管理　进程、核级线程、用户级线程是三种范例，但是都不尽人

意。协调型用户级线程模型是取得性能、灵活性、正确性三者统一的最佳方式。并行表达、用户级与核级信息流动方式是探索的核心。

(2) 操作系统的构造生成　微核相对于 hosted system、Symmetric Sysrem 具有较强的优势，它也是从 PC、工作站到并行机 OS 发展的方向，但是 MPP 的微核在功能划分、构造方式上需继续研究，它应以支持通讯为核心来构造，而将服务界面放在第二位。

(3)MPP 的节点结构　合理的软件设计是重要的，在 MPP 的结点结构设计上如何消减、屏蔽 OS 的影响更是至关重要。纵观最近的 20 多种高性能 MPP 机，它们的不同之处不仅直观体现在互连结构上的差异，同时结点结构设计也各具千秋。节点结构上的不同反映了不同探索减少软件延迟开销的研究，OS 作为对应用程序的支持系统，本身存在着与应用程序的物理并行性。开发这种并行性、探索合适的节点结构是提高 MPP 的性能的重要方面。

(4) 微处理器结构　近期计算机的十余年的发展，最明显的莫过于微处理器突飞猛进的发展，微处理器的发展使其功能越来越强大，集成的功能部件也越来越多，但是这种引用以前巨型机结构的设计方法是否有利于并行的进一步开发值得深究，因为众多的寄存器，深度流水线和大量的并发单元形成了执行单元的现场，执行单元的切换引入了大量现场的保存和恢复时间开销。微处理器结构设计同时应考虑 OS，二者不应是独立的发展。同时将来的微处理器上也许本身就是多 CPU，CPU 之间如何协调工作值得研究。

2　增强互连网络的可伸缩性

互连网络是 MPP 的主要组成部分，它占据了系统总代价的很大一部分。现阶段互连技术所取得的进步主要归功于流控制技术的发展，W. J. Dally 和 C. L. Seitz 在 1986 年提出的 Wormhole 技术是目前互连技术所取得的最大成就之一。它具有低网络延迟、通道共享性好、易于用 IC 实现、易于实现 multicast 和 broadcast 等通讯，Wormhole 与虚通道技术的结合，改善了互连网络的延时和其资源的使用效率，使得延迟呈现出与传输路径长度的不敏感性。正是因为如此，有些文献认为拓扑结构不重要，互连网络无须研究，这些看法只是注意到了互连网络的静态属性（直径），而忽视了其伸缩性和动态属性（阻塞延迟）两个重要特性。

可伸缩性是表征增大机器规模的难易程序和系统规模的增大对性能影响程度的一种定性参数。动态属性反映在消息在传输过程中被其它消息所阻塞的延迟时间上。互连网络的这两种属性一方面与工程实现（包装技术）密切相关，同时又受应用问题所产生的消息流量和通讯方式的影响。包装工艺技术的难以抽象、应用问题属性的多种多样，使得定量化的描述这两个参数十分困难。

MPP 所采用的基本的互连网络可分成 k-ary n-cube（如 MESH、Torus ）和 k-ary nfly（如 Omege 、banyan 、Delta）两大类。这两类互连网络都可用于 MPP，各自存在着优缺点。k-ary n-cube 网络是 hypercube 的改进，削弱了由于节点度限制带来机器规模难以扩展的问题，保留了结点的对称性、路由算法简单的优点，而 k-ary n-fly 则与之相反。

目前 MPP 的互连网络结构主要沿着三条途径发展：一种是低维的 k-ary n-cube 网络（如 2D/3DMESH 、2D/3DTorus 等），如 iPSC/l、PargonXP/s 、AP1000、Alewife 等。采

用这类网络的机器，其规模扩展容易，但是要保持性能上的可扩展性，十分困难，例如对 MESH 网络而言，随着处理机个数的增加，若要维持每一结点的通信带宽，信道宽度需以 $O(\sqrt{N})$ 增长。另一种是多级互连网络和 Crossbar，由于它们具有很高的带宽，随着工艺技术及包装技术的发展，也被用于较大规模的 MPP 机，如 IBMSPx、NEC Conju-3、Meiko CS-2 等。随着机器规模的增长，这类互连网络的通讯能力也逐渐增强，其消息饱和量维持为 $1/W$（W 为信道宽度），但是机器规模的增长却缺乏灵活性，处理机个数只能以前规模的 K 倍增长。第三种为层次互连网络，如 CCC（Cyclic connectedCube）、多级总线，以及目前较多的 k-ary n-cube Cluster-c。层次互连网络 Cluster-c 其低层 C 个处理机采用总线或互连网络构成，K^n 个这样的 Cluster 互连成一个 k-ary n-cube 网络，如 Intel Paragon、Stanford DASH 等。针对互连网络结构，有以下两个方面需进行研究：

（1）MPP 的互连网络设计应基于包装技术，面向可伸缩性，减少传输的阻塞延迟

MPP 机的组装本身隐含着一种层次属性，它通常需几级组装（如机柜、机箱、印制板、VLSI 芯片等），不同包装级的互连密度、网络代价、速度各不一样，尽管模型化包装技术十分困难，但是包装中的一个基本特征是：低层的连接比高层的连接更加便宜、密度更深、速度更快，因此采用层次式的互连网络是必然的。同时通讯局部性是大规模并行程序、算法中普遍存在的现象，层次式互连网络可开发这种通讯局部性，减少通讯延时。

（2）在通讯库中对消息传递进行并行调度

宏观上看一个处理机发送何种消息，何时发送对另一个处理机而言是未知的，这使得对所有处理机所发消息进行并行调度是不可能的。但是存在一些需要多种消息或多次发送的通讯方式需在一组处理机间进行完成，这种通讯方式称为消息密集型通讯（Collective Communication），这种通讯方式和这组处理机对程序员都是可见的，在通讯过程中有时它们存在固有的路径冲突，因此在通讯库中采用并行算法的思想对它们进行调度以消除冲突是至关重要的。

3 设计高性能路由器

影响通讯中的网络基本延迟和阻塞延迟的另一结构部件为路由器。MPP 十多年的发展使得路由功能从以前采用软件实现发展到普遍采用独立的部件实现。同时流控制的发展也使得计算机网络中的路由器功能也可采用 ATM 开关实现，目前无论是在 MPP 的互连网络中，还是在计算机网络中，路由开关的设计方法研究都十分热门。

Wormhole 流控制算法从原理上讲，在每一个开关的输入输出端口仅需 1 个 Flit 的缓冲区，然而消息一旦被阻塞，由于它需保留所占有的信道，所以在路由器的 buffer 空间很小时，会导致较大的网络阻塞延迟，并且可能会导致死锁，为了减少消息被阻塞的概率，避免死锁，人们又提出了虚通道（Virtual Channel）和虚跨步（Virtual Cut-through）的流控算法。这两种方法都需路由器中具有较多的缓冲空间和有效的缓冲空间管理。探索路由器中关于这些方面的结构设计，显得必要。目前的路由器设计中缓冲区的实现一般采用寄存器，采用寄存器的设计方法带来了路由器功耗高、缓冲区器空间受限等问题。采用存储器来替换寄存器是一种有效的方法，因为其功耗小、逻辑简单而且容量可以做得

较大。VLSI 路由器主要不是受其逻辑门数所限，而是引脚数起决定作用，存储器的采用可有效的利用硅片面积，且片内存储器的访问速度与寄存器的速度相差无几，在受限的网络频率下，它们的访问时间相当。同时由于消息密集型通讯的消息流量大、通讯时间长，我们也提出了一种采用这种结构的路由器实现 multicast 通讯方式的有效方法[5]，修改了传统基于发送或基于接收的 multicast 协议，分析说明这种方法能取得较高的效率。

路由算法是路由器设计中需研究的第二个重要问题。路由算法可分为确定性（deterministic）和自适应（adaptive）两大类，确定性由于算法简单而易于实现，但是一旦消息被阻塞时，无选择其它空闲路径的能力。自适应路由算法消除了这种缺陷，能有效地减小网络的阻塞延迟，提高网络利用率。然而死锁（deadlock）与活锁（livelock）是须避免的两个重要问题。自适应路由算法的实现需要更多的逻辑功能，其实现困难。Chaos 路由器是第一个提出自适应路由器算法实现的路由器，但由于物理实现上的困难性，并未投入应用。自适应算法的简洁性是投入应用的关键。Cray T3E 中的路由器设计迈出了一大步，使得其具有自适应的寻径能力。

提高互连网络传输速度的最直接、最有效的方法是提高其工作频率，但这并非易事。目前的多数 MPP 采用强同步的工作方式，即互连网络的工作频率由唯一的振荡器提供，每一个路由器与相邻路由器的时钟必须严格对准，且在相邻路由器的通讯链路上，数据必须在一拍时钟周期内完成，这种方法使得网络频率的提高十分困难，布线长度是最直接的原因之一，网络的工作频率必须低于系统中最长线的延迟。解决此问题的途径是通讯链路采用流水线式的传输方式，这种链路称为 Pipeline Channel，此时每条链路上不再是唯一的一位数据传输，而可能是多位数据。这种设计方法突破了网络工作频率受限于系统中通讯链路布线长度的无关性。Pipeline Channel 的采用，需要探索路由器之间的传输协议，保证数据传输能被正确的接受。

4 结 论

本文在分析产生通讯延迟的几个主要因素的基础上，提出了消减 OS 的影响、增加互连网络的可伸缩性、设计高性能路由器等提高 MPP 通讯速度的方法。在每一部分中阐述了研究背景，并提出了一些亟待研究的问题。

参 考 文 献

1 Anderson T E et al. Scheduler Activations; Effective Kernel Support for the User-level Management of Parallism. OS Review, 1991, 3

2 Giloi W K et al, MANNA: Prototype of a Distributed Memory Architecture With Maximized Sustained Performance. Tcehnical Report

3 Raghunath M T, Rande A. Design Interconnection Networks for Multilevel Packaging. The Superconputing 93.

4 Cheng C H, Mchinley P K. Communication Issues in Parallel computing Across ATM Networks. IEEE Parallel & Distrbuted Technology, Winter 1994

5 Xiao L Q, Zhang M X, Zhou X M. A multicast Protocol in Multistage Interxconnedtion Networks. International Symposium on Parallel Archtecture, Algorithms and Networks, IEEE computer society Press, 1996

（责任编辑 张 静）