PM：高性能集群计算通信库

伍　红，黄传河，刘晓明，江　贝
(武汉大学 计算机技术科学学院，湖北 武汉 430072)

摘　要：介绍了一个高性能的通信库：PM。PM是用Myrinet千兆局域网卡连起来的工作站集群。网卡上有一个专门的处理器和内置式存储器，用于处理通讯协议。PM实现了高性能通信和支持多用户环境。它是一个操作系统，通过Deamon进程和编程语言的运行时间例程实现。PM的独特特性：网络内容交换和改进的应答/否定应答(ACK/NACK)流控算法等。太阳机上的PM已达到一个用户级8字节消息全程传送需20μs，8K字节消息的带宽为38.6M字节/秒。

关键词：集群计算；千兆网；用户级通讯协议；高性能计算
中图分类号：TP393　　**文献标识码**：A　　**文章编号**：1001-3695(2000)12-0035-03

1　引言

最近几年，两个重要的趋势：全局计算和集群计算，已经吸引了很多人在高性能网络计算方面的兴趣。

使用高速互连技术，如ATM、光纤通道和Myrinet，集群高性能商业工作站和PC机组建并行计算平台，已经成为可能。这些互连技术，“Killer交换机”具有大规模并行处理(MPP)互连的性能和可扩展性，也具有局域网的灵活性。只要简单把这些交换机和最新高性能商业部件互连在一起，就能组建一个速度快的并行系统。

PM是一个高性能工作站集群的通信库，通过Myrinet千兆网卡相连。PM使用的编程语言叫MPC++；操作系统叫SCore，它支持多用户并行处理环境。SCore是通过一个被称为SCore-D的Deamon进程实现，它处于Unix操作系统上层。SCore管理工作站上的一系列被称为并行进程的用户进程。SCore-D应用群调度改变并行进程的内容，包括它的网络状态。它允许多用户按时空共享(TSSS)的方式访问工作站集群。PM提供一种可靠的FIFO序的消息传递机制，这也是MPC++运行时间例程的前提假设。

Myrinet网络接口有一个专门的处理器，它允许用户设计自己的通信协议。为达到低延时、高带宽通信，用户存储器映射的网络驱动器被广泛运用，例如：活动消息(AM)、快速消息(FM)和UNet。在这些驱动器中，用户进程直接访问网络硬件，以减少核心陷入和数据复制。然而，因为进程排它地使用网络硬件资源，大多数通信设备在一个时间只能被一个进程使用。PM支持通信通道，它代表了网络状态和包括内置存储器上的消息缓冲区。SCore群调度内容交换机制支持多通信路径。改进的应答/否定应答流控算法和立即发送技术的提出，提供了高吞吐量、低延时通信，保证了FIFO序消息传递。

收稿日期：2000-06-07

2　PM：等同于操作系统的高性能通信库

2.1　工作站集群和Myrinet千兆局域网卡

PM是一个Myrinet上高性能通信库。Myrinet是Myricom公司制造的商业千兆局域网。图1表明了一个工作站集群的结构，它是由36台Sun Sparc Station 20s通过Myrinet连接起来的。

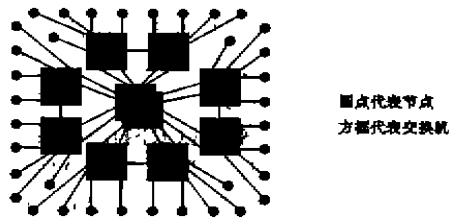

图1　网络拓扑结构

Myrinet局域网接口卡有一个32位的微处理器，一个静态随机存储器(SRAM)，一个网络接口和一个SBus或PCI DMA控制器。内置式处理器执行存储在SRAM中的程序，去控制网络接口和DMA控制器。因此，我们可以通过改变卡中的程序开发Myrinet网上新协议。

因为链接上的所有消息传递或接收都必须先保存在SRAM中，我们需要SRAM与主存之间的专门数据传送。内置式SRAM位于SBus或PCI地址空间。通过映射SRAM地址域到用户的地址空间，用户可以直接存取在SRAM中的数据。低延时消息传送使用终端设备定时询问技术(Polling)。

Myrinet不同于其它局域网硬件，如以太网和ATM。区别如下：(1)使用硬件流控保证消息传送；(2)通过相同路径传送维护消息的次序。

2.2　编程语言和操作系统

MPC++编程模式是基于多线程的，运行于分布式存储器的并行机。除了等待同步或离开，线程继续执

行。MPC++第二版模板库，被称为多线程模板库(MTTL)，支持本地/远程线程援引，同步结构和全局指针。MPC++运行时间系统处理通讯和用户级线程，不需要中断驱动的通讯机制。MPC++运行时间系统以可靠的FIFO序消息传递机制为前提。MPC++编程模式是基于分布式存储器的并行机，远程线程通讯要求低延时、高吞吐量。

工作站集群的操作系统被设计成守护神(Daemon)进程，称为SCore-D。它通过Myrinet与其它SCore-D守护神进程通讯，实现群调度。也就是说，用户和守护神进程同时存取Myrinet硬件资源。为了让多个进程(SCore-D和用户进程)同时使用通信设备，通过通道实现连接，消息从发送节点的通道传送到接收节点的通道，这两个通道要相同。每个进程占用一个通道。通道必须有自己的消息缓冲区，以便一个通道拥挤不会影响其它通道。

因为局域网卡内置式存储器资源的限制，不可能根据需要建立任意多的通道。一个通道必须被多个进程分时使用。实现这一点，需要通道内容交换机制。通道内容包含与通道有关的所有存储器，包括消息缓冲区和在网络中流动的消息。

2.3 PM协议的设计

PM采用终端设备定时询问技术，减少通讯时延。在PM中，局域网卡的内置处理器传送消息，用户进程的接收线程定时查询到来的消息。终端设备的定时询问立即通知接收线程关于消息的到来，使低延时通讯成为可能。普通局域网(如以太网)使用系统调用和中断，并不能满足低延时的要求。通过ATM连接Sun工作站上的AM(活动存储器)(称为SSAM)，使用特殊的陷阱指令去解决这个问题，需要对操作系统内核作修改。

因为Myrinet网络只通过内置式SRAM存取数据，数据在每个节点宿主机存储器之间传送有如下四个步骤：(1)通过DMA从主存储器传送数据到内置式SRAM；(2)从内置式SRAM发送数据到网络；(3)从网络接收数据到内置式SRAM；(4)从内置式SRAM传输数据到主存。

DAM和消息传送的顺序执行不能充分利用DMA控制器和网络接口的带宽。尽管双缓冲区允许当前消息正在传送时DMA发送下一个消息，它增加了吞吐量，但并没有减少每个消息传送的时延。

PM开发了一个新技术，称为立即发送。在DMA从主存传送数据到SRAM开始之后，马上开始从SRAM发送数据到网络，像流水线处理。立即发送不仅可以增加吞吐量，而且可以减少时延。

在接收节点，接收一个消息和提交DMA传送同时进行是不可能的。因为在整个消息被接收之后，将检测CRC错误效验码。PM在接收节点使用双缓冲技术去提高吞吐量。

如果宿主机程序不接收正在到达的消息，可能是因为内置式存储器上的消息缓冲区溢出。PM设计了一个新的基于应答/否定应答协议的流控算法。

工作站集群的流控方法甚至在很多节点的情况下也应具有伸缩性。基于窗口的流控算法不具有伸缩性，因为接收节点必须管理每个发送节点专门的接收缓冲区，大量的节点使有效缓冲区大小变得很小。普通的“应答/否定应答和重传送”方法或“返回发送方”方法没有这种分离缓冲区的问题，但它们把不能被接收的消息送回发送方，也不保存消息的次序。

PM已设计出既保存消息次序又具伸缩性的流控算法。这种算法被称为改进的应答/否定应答流控算法。它使用应答/否定应答和重传的思想，引入发送方/接收方的状态来保存消息的次序。在这个算法中，保存消息的次序，是因为连续发送的消息并没有被接收，直到能初始接收的消息才被正式接收。这种算法要求发送方消息缓冲区存储消息。当发送方收到这个消息的应答消息后，才可释放这块消息缓冲区。这个缓冲区并不占用很大空间。Myrinet硬件保证消息的传送，允许我们使用这种简单的流控算法。

改进的应答/否定应答流控有两个优点：一是发送节点知道消息是否被接收节点接收。我们利用这个特性实现通道内容交换。PM通道的内容包括数据结构和与通道有关的、在宿主机存储器和内置SRAM上的缓冲区。为交换通道内容，这些存储器域被保存到宿主机存储器上，下一通道内容被加载。通道内容交换必须不会引起消息复制、消息丢失或消息混合。为保证这一点，只有当没有正在出去的消息或正在到来的应答时，通道内容才可以交换。使用PM的流控算法，接收所有已发送消息的应答/否定应答信号，保证了没有消息存留在网络上；另一个优点是不管接收消息缓冲区是否溢出，所有消息没有堵塞的被发送到目的地。如果一个用户进程通过一个通道发送大量消息，堵塞了网络，别的通道也无法发送消息。在这种情况下，使用通道的SCore-D操作系统，也不能进行。改进的应答/否定应答流控会预防这种情况的发生。

PM流控算法的不足是当重传时，可能增加网络负载，因为发送方不断发送消息，直到否定应答信号到达发送方节点。

2.4 性能

在两台通过Myrinet交换机和Myrinet LANai4.0硬件相连的Sun Sparc Station Model 20/71上测量了PM的基本性能。图2显示了PM的全程时间。在这次评价中，改变消息的大小，从8字节到8 192字节，测量了：(1)使用立即发送；(2)不使用立即发送。8字节消息的PM全程时间大约为20μs。图2表明了对大消息来说立即发送技术减少了时延。

图2显示了PM的吞吐量。在这次评价中，改变消息的大小，从8字节到8 192字节，测量了：(1)使用立

即发送和接收双缓冲；(2)只使用立即发送；(3)不使用立即发送和接收双缓冲。图2表明了立即发送技术提高大消息的吞吐量，接收双缓冲提高所有消息的吞吐量。8 192字节消息的PM最大吞吐量是36.8M字节/秒。

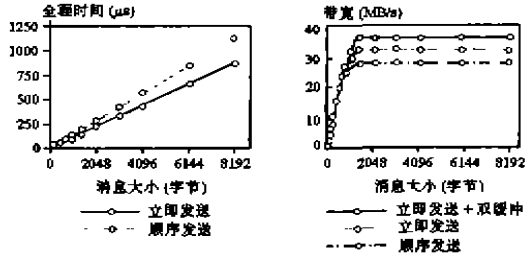

图 2　全程时间和吞吐量

表1显示了保存和加载通道内容的时间。测量了四种情况：(1)缓冲区没有消息；(2)发送缓冲区满：有511个消息(12K字节)；(3)接收缓冲区满：有2 730个消息(32K字节)；(4)接收和发送缓冲区都满。

表1 内容交换时间(ms)

条件	保存	加载
没有消息	0.13	0.11
发送缓冲区满	1.88	1.40
接收缓冲区满	3.39	1.95
发送和接收缓冲区满	5.15	3.22

当有消息在缓冲区时，保存和加载通道内容需要更长时间。这是因为PM的消息缓冲区相当大，通过SBus存取内置SRAM的开支也很大。显然在保存和加载内容时，转移同样数目的数据，保存内容占用更长时间。这是因为从SBus空间读数据比写数据花费更多的时间。

3　总结

本文展现了高级网络计算中的一个重要的通信库：高性能集群计算的PM。PM是通过Myrinet连接商业工作站和PC机。本文评价了PM库的性能。结果表明工作站集群达到了相同数目处理器的Cray T3D的性能。PM允许用户开发现代高性能处理器的并行系统，并在SCore操作系统下共享系统。

参考文献：

[1] P Arbenz, W Gander, M Oettli. The Remote Computational System[J]. in: High-performance Computation and Network, Lecture Notes in Computer Science, Springer Berlin, 1996, 1067: 662-667.

[2] N J Boden, D Cohen, R E Felderman, A E Kulawik, C L Seitz, J N Seizovic, W K Su. Myrinet- A Gigabit-per-second Local-area Network[J]. J.IEEE MICRO, 1995, 15(1): 29-36.

[3] H Casanova, J Dongarra. NetSolve: A Network Server for Solving Computational Science Problems[R]. Technical Report University of Tennessee, 1996.

[4] A S Grimshaw, W A Wulf, J C French, A C Weaver, P F Reynolds Jr. Legion: The Next Logical Step toward a Nationwide Virtual Computer[R]. Technical Report, University of Virginia, 1994.

[5] A Hori, H Tezuka, Y Ishikawa, N Soda, H Konaka, M Maeda. Implementation of Gang-scheduling on Workstation Cluster[Z], in: D G Feitelson, L Rudolph (Eds), IPPS'96 Workshop on Job Scheduling Strategies for Parallel Processing, Lecture Notes in Computer Science, Springer, Berlin, 1996, 1162: 76-83.

[6] Y Ishikawa. Multi Thread Template Library-MPC++ Version 2.0 Level 0 Document[R]. Technical Report, TR-96012, RWC, 1996.

[7] Y Ishikawa, A Hori, H Tezuka, M Matsuda, H Konaka, M Maeda, T Tomokiyo, J Nolte. MPC++[Z]. in: G V Wilson Paul Lu (Eds), Parallel Programming Using C++, MIT Press, Cambrige, 1996: 429-464.

(上接第29页)直传事务号n；然后申请一遍内存区(当无空间可用时可自行分配根本不存在的一遍虚空间)，该内存的始地作为直传事务的始地DTn_LB；根据其它节点的读请求得知DTn_LN、DTn_RB、DTn_RP等参数，通过NI的DRIVER和这些参数启动直传打包部件；第四步启动读盘，将盘数据读入该内存区中。这样直传部件析每检测接收到一个直传数据就打包发送，长度寄存器减一；直至长度寄存器减为0， NI向CPU发中断。最后是CPU的中断服务和必要的通讯。

上述的通讯软件是在用户级做的。在当今的Linux环境下，由于源代码公开，对特定MPP系统可考虑将它做在核心级，重新构建具有直传功能的内核。其工作原理同上类似。

3.4　直传的性能分析

以上的描述中都是采用的单PCI总线，具体实现时可以采用两条或多条PCI总线(只要工程实现允许)。

直传的性能：假设PCI总线条数为N，每条PCI总线传输率为R，NI和主存的传输率分别为R_{NI}、R_{MEM}，数据大小为S_{data}。对于必须进行磁盘读的文件I/O请求，本系统中两种方式下的性能差别：

直传方式：$t=S_{data}/(N\times R)$

非直传方式：数据先进入内存：　$t_1=S_{data}/(N\times R)=t$ (受限于PCI总线的传输率)

数据从内存进入计算节点：$t_2=S_{data}/R_{NI}$ (受限于NI的传输率)

加速比：$(t_1+T_{os}+t_2)/t = 1+(T_{os}+t_2)/t>1+(N\times R)/R_{NI}>1$

显然，当CPU处理速度慢时，T_{OS}增大；当PCI总线传输率总和(N×R)同网络接口部件NI传输率相接近时，t_2/t增大。这两种情况下，采用直传是很有意义的。

4　结束语

当今高性能I/O倍受人们关注。在多流并发I/O节点逻辑实现技术基础上，本文重点讨论了数据从盘设备到互联网的直传。下一步的工作主要进行该硬件系统的DRIVER及通讯软件的开发和进行优化。

参考文献：

[1] PCI Special Interest Group. PCI Local Bus Specification[M]. revision 2.1, JUNE 1, 1995.

[2] 李东晖，两条PCI多流并发I/O结点实现技术研究[J]. 计算机应用研究, 2000, 17(6): 4-7.