来越多关注的一项重要应用是数据库中的知识发现(KDD)。知识发现或数据库的挖掘是人工智能的一个相对新的子领域，它涉及到从不断增长的企业信息数据库中挖掘出额外的非平凡的知识方面。在这个内容上，主要任务之一是内部数据的关联或关系的发现和特征；例如，医学数据库中症状和病症之间，出现在这种关系中的基本因素的揭示描述将帮助用户更好地理解关于被收集数据的现象的本质，或者说它能被用来进行预测[20,21,24,25]。另一方面，运用 Rough 集方法，还能处理的另一方面是对数据中的反常模式或行为的发现，目的是为了检测假的或非法进入的数据[23]。

KDD 的方法学主要源于早先统计学、数据库理论和机器学习领域中的研究[22]。然而，Rough 集领域中的更新发展确立了这一方法学成为 KDD 问题的主要方法之一[23-25]。

Rough 集技术被用于相关 KDD 研究目的已经五年了。特别是，对数据库的挖掘，比如说 Datalogic[20]，基于 PC 的商业软件系统的有效性使得这种技术从工业和科学的不同部门的用户都是有益的。当前，Rough 集方法学正在其它领域中被引用：市场研究[25]、医学数据分析[17]、药物研究[8]、以控制为目的的传感器数据分析、以及导致新的合成材料设计的研究。库存市场数据的分析已证明了一些著名的市场规则，并导致了一些新的重要规则的发现[25]。利用 Rough 集原始模型扩充的知识发现方法学称为可变精度 Rough 集，Regian 大学在最新的基于工作站的工具集中已经实现了关于知识发现的决策矩阵方法[23]，称其为 KDD-R，KDD-R 被用来对医学数据分析并且当前正支持电信工业的市场研究。

结论 Rough 集方法学已经证明了它在许多实际生活应用中是完备的和十分有用的。Rough 集理论提供了在 AI 的许多分枝上可应用的有效方法。Rough 集理论的有利条件之一是实现它的方法的程序可以很容易地在并行计算机上运行。

然而，许多问题仍尚待解决。尽管 Rough 集理论产生于真正的数学基础，但许多理论问题仍有待于真正澄清。Rough 逻辑，一种基于 Rough 集哲学的不精确推理的逻辑，它似乎是最重要的课题。基于 Rough 集理论对神经网络和遗传算法方法的开发也似乎是很重要的。Rough 控制，即基于 Rough 集理论的控制也似乎是一个非常有前途的应用领域。然而，基于 Rough 集哲学的一个定性的控制理论必须被建立。Rough 集理论与非标准分析、非参数统计学及定性物理学之间的关系是另一些重要课题。

致谢 作者在此对 T. Y. Lin 的帮助和鼓励表示谢意。

译自《Comm. of the ACM》,38(11)1995,PP94-95)

(上接第 37 页)

[6]Thekkath & H. M. Levy, Limits to Low-Latency Communication on High-Speed Networks, ACM Trans. Comput. Syst., 11(2)1993

[7]T. von Eicken et al., Active Messages: A Mechanism for Integrated Communication and Computation, in Proc. 19th Ann. Int. Symp. Comput. Architecture, May 1992

[8]Tucker and A. Mainwaring, CMMD: Active Messages on the CM-5, Parallel Computing, 20(4)1994

[9]Oliver A. Mcbryan, An Overview of Message Passing Environments, same to [8]

[10]王鼎兴、庄伟强，一种实现并行计算的新主流技术—NOW，小型微型计算机系统，16(2)1995

[11]Ronald J. Vetter, ATM Concepts, Architectures, and Protocols, CACM, 38(2)1995

[12]MPI Forum, MPI: A Message-Passing Interface Standard, May, 1994

[13]Al Geist, et al., PVM: A Users' Guide and Tutorial for Networked Parallel Computing, 1994

计算机科学1997 Vol.24 №.1

NOW环境下并行计算中的通信时延问题

吴礼发 谢 立 孙钟秀
(南京大学计算机系 南京210093)

摘 要 This paper analyses all the factors leading to communication latency in parallel computing in NOW, and discusses the methods to reduce the latency.

关键词 NOW, Parallel computing, ATM, Active message.

1 引言

工作站网络(NOW,Network Of Workstations)是一组专用或通用的计算机特别是高性能工作站通过网络连成的计算机系统,由于NOW用于并行计算的主要资源是工作站,所以它又被称为工作站群(Workstation Cluster)。虽然将NOW用于并行计算的概念已出现了许多年,但是只有近几年因高性能工作站和高速网络如ATM的出现以及分布环境中支持并行计算的软件开发环境的发展才迅速发展起来。在NOW之前主要有三种并行计算系统,第一类为多向量处理机系统,主要用于大型科学计算和工程设计;第二类为基于共享存储器的多处理机系统,可用作服务器并同时进行并行处理;第三类为基于分布存储器的大规模并行处理(MPP)系统,主要用于解决要求极高运算性能的影响国计民生的重大挑战性问题。由于费用昂贵和软件及编程环境的不完善使得这三类系统不能得到推广,同上述三种系统相比,NOW具有用户投资风险小,用户编程方便,系统结构灵活,性能可随着工作站性能的提高而迅速提高,能充分利用现有工作站资源等优点。

一个实用的NOW还应有一个高效的软件环境。支持NOW进行并行计算的软件结构如图1所示。NOW的操作系统一般为通用的UNIX、AIX等,为了支持某些特殊的操作,也可以对操作系统进行修改和重建;通信协议实现工作站到互连网络的数据通信服务;通信原语库由用户编程调用;并行程序设计环境和并行工具包则提供用户方便、友好的使用接口。

MPP领先于NOW技术的一个主要方面是互连通信网络,但是随着HPS,Gigaswitch等专用高速网,尤其是ATM局域网的出现,这种优势已逐渐消失。然而,如何更有效地利用高速网的高带宽、低时延的优点以减小目前存在的性能差异仍然是一个很重要的研究课题。

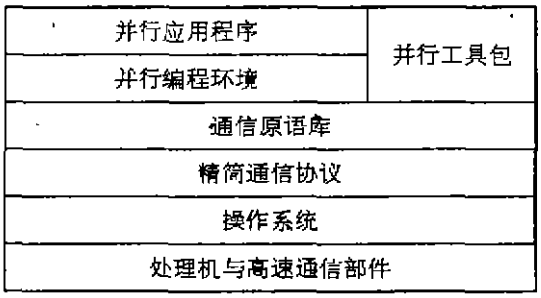


图1 NOW的软件结构

从图1中我们可以看出,在NOW环境中的通信时延与通信网络、操作系统、并行计算环境等有着密切的关系。因此,本文对NOW环境中并行计算中的通信时延作了比较详细的分析,并讨论了解决通信时延问题的方法。

2 通信时延分析

2.1 通信时延

一般来说,通信时延由两大部分组成:硬件时延和软件时延。前者是由信号在物理链路上的传播时间和网络控制卡所带来的时延,网络控制卡时延指从发送方控制卡可以从主机得到数据到开始数据传输到网络链路上的时间或从数据到达接收方控制卡开始至接收方主机可以得到数据的时间。软件时延源一般由以下三部分组成[6]:

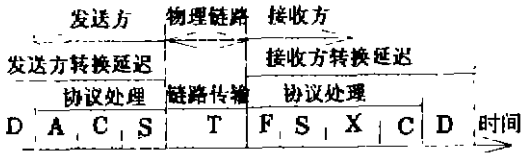
·操作系统。在许多系统中,进程间通信需要操作系统的干预,由操作系统来负责通信资源的保护和管理。调用系统调用所带来的时延往往是总的通信时延的很大一部分。同样,因处理中断所进行的上下行文切换也是通信时延的一个重要组成部分。在

用户态和核心态之间切换的次数越多，时延将迅速增长。

·编程接口。报文传递系统一般都要提供许多种通信方法。应用接口可能在传送报文前还进行扩展的缓冲区管理、检错、上下文检查、地址重映象等工作。接口的复杂性也是软件时延的一个重要组成部分。

·协议。特殊的报文传递协议的语义可能要求接收进程在接收其它进程传送来的报文前检查呼叫请求的报文标志，分配和管理缓冲区以及执行其它的任务。一次或多次握手需要执行。在大多数系统中，这些功能都要求在数据被传送之前完成。

将一次通信时延进一步划分，则如图2所示：



D 为操作系统调度时间；A 用于分配缓冲；
X 用于从接收队列获取数据；
C 用于拷贝数据到系统缓冲区；S 用于启动接收/发送；
T 用于链路传输；F 用于定位中断源。

图2 一次通信时延

从上面的分析可以看出，数据拷贝（涉及到缓冲区管理）和上下文切换是通信时延的重要来源。对于发送一些小的报文或甚至是几百个字节长的报文，上述的软件开销所带来的时延太大。因此，就产生了这样一个问题：是低层的硬件或系统软件为应用提供更多的报文传递支持还是应该由程序员或编译器来决定是否以及什么时候采用一定的报文传递策略。作者认为，由报文传递系统提供高效的进程间通信支持，由程序员来决定给某一应用选择合适的通信方法是比较有效的方法，因为不同的应用有着不同的通信需要和不同的通信特点，因此，将通信策略交给程序员来决定是合适的。

2.2 通信网络对通信时延的影响

NOW 的基础是通信网络，因此连接 NOW 中的各计算机的通信网络的性能好坏直接关系 NOW 并行计算系统的性能。

以太网是一种以10Mb/s 速率采用 CSMA/CD 多路访问方式的流行局域网。其网络拓扑结构是总线型，网络各站点以随机发送、冲突重发的方式共享传输媒体，因此在低负载下具有比较高的效率。但在重载下，因为冲突的机率增大，网络效率将大大降低。

快速以太网是近几年来发展起来的一种新型的以太网，其传输速率为100Mb/s，是传统以太网的10倍。

FDDI（光纤分布式数据接口）是一种光纤令牌环网络，其传输速率为100Mb/s。它一般采用双环拓扑结构。网络站点接在环上。令牌在环上移动，各站点只有在收到令牌后才能发送数据。当环上结点增多时，网络性能将不断下降。当业务量较小时，令牌传递所带来的时延比较大，网络效率不高。尽管 FDDI 提供较高的带宽，但是综合起来看，令牌环并不适合低时延通信。

异步传输模式（ATM，Asynchronous Transfer Mode）是为宽带综合业务数据网（B-ISDN）的网络层制定的一个标准，是一种快速的分组交换，采用固定大小的数据单元：信元（cell）。信元的长度为53个字节，其中，5字节用作头信息（包括标志、信元丢弃优先级、路由和交换信息等），48个字节用作数据域。ATM 是基于交换的通信网络。同基于共享媒体如同轴电缆或 Token Ring 等传统的局域网相比，它可以提供巨大的总带宽：多个分组可以同时以全通道速率通过交换机。这为有效地实现并行计算中的群通信提供了很好的通信基础。

通信网络所带来的通信时延主要表现在缓冲区管理、错误处理、中断处理等。比较上述四种网络，以太网的带宽最低，ATM 网络的带宽最高。总的来说，基于交换的网络如交换式以太网、ATM 网络等所带来的时延比较小，且 ATM 网络将纠错和重传在端端实现，有利于降低通信时延。一些通信网络协议如 TCP/IP 比较复杂，因而有较大的通信时延，不适合并行计算的通信。对于大的数据传输，FDDI 的性能优于 ATM 网络[5]，但 FDDI 的令牌环使得它不适合进行低时延通信。

综合起来看，ATM 局域网的高带宽低时延的特点使得在 ATM 网络上进行并行计算是一个主流方向。目前主要在 ATM 局域网上作并行计算，将来可以很容易地将现有的应用移植到 ATM 广域网上作并行计算。

2.3 消息传递系统对通信时延的影响

因为在 NOW 环境中编写并行应用程序是在消息传递环境中进行的。消息传递环境的效率直接关系到并行系统的效率。下面我们对几种比较流行的消息传递环境在通信方面的性能作一个简单的说

明。

RPROC 系统是由 McBryan 开发的报文传递系统，具有很多当前异构的报文传递系统的特点，特别是异构性、主动消息、混合数据分组和自动数据类型转换。该系统的目的是将运行不同操作系统的计算机互连起来。RPROC 支持异步通信和计算与通信重叠，RPROC 报文是真正的主动消息，在报文头部有接收方执行的中断子程序的地址，因此有利于提高并行系统的效率。它能运行在10多种体系结构上，可以构造一些大的异构环境中的应用。

P4系统是由 Argonne 国家实验室开发的报文传递系统，是最早开发的可移植性平台。它支持共享内存模型和分布式存储模型(用消息传递)。它是一种效率比较高的系统，主要因为它是一组宏而不是一组函数，省去了函数调用所带来的时延。它也支持进程管理。其主要缺点是消息传递是阻塞的。

Express 是美国 Parasoft 公司推出的能在不同硬件环境上运行的并行程序设计环境。它强调并行程序设计的高层问题，目的在于更进一步为用户提供方便的并行程序开发工具。在开始的时候，Parasoft 公司在许多体系结构上实现 Express，并且想在每一种体系结构上都获得较高的性能。这一做法还是比较成功的，常常降低通信时延4个或更高的数量级。Express 的核心是一组关于通信、I/O、并行图的函数库。在通信方面，Express 提供了多种处理机间通信原语，除了通常的同步/异步通信之外，同时还提供了广播及多重接收等功能函数。

PVM(Parallel Virtual Machine)是一个在异构型网络环境中模拟一个通用的分布式存储多处理机的软件系统。它是网络计算的一个支撑软件。有了它，就可以在即使不具备真正的多处理机的情况下进行并行计算。由于它的适应性(PVM 的虚拟并行机可由任意不同种类的机器组成)和它简单但完全的编程界面，PVM 系统在高性能科学计算中获得广泛的应用。PVM 报文传递原语是面向异构的操作，能够处理与缓冲和传输相关的强类型问题。在通信方面，不仅支持一般的发送和接收操作，还支持高级通信原语如广播、路障同步、全局求和等。

Linda 是由 Yale 大学的研究工程中演化而来的一种并行编程模型。其最基本的概念是 Tuple 空间，它是协调进程间通信方式的一种抽象，是一个程序语言提供的虚拟存储模型，其基本特点是共享性、同步性和关联性。它的最大特点在于其并行程序设计机制是通过扩展标准的程序设计语言(加入支持并行通信功能)而得到的，如 C 和 Fortran 语言。像这样开发的并行共享内存应用程序实际上只有在分布内存的并行系统上实现运行，其中应用程序各并行进程间的报文传递都通过 Linda 翻译器来自动实现。

Linda 对 Tuple 空间的操作提供了进程管理、同步控制以及通信函数等 MIMD 并行程序所需的基本并行操作。

报文传递接口标准 MPI(Message Passing Interface)的正式说明是1994年4月完成的，是报文传递系统发展到一定阶段后，人们普遍希望定义一种消息传递核心库函数的语法和语义以满足更多的用户的需要和在更多的平台上运行的结果。定义 MPI 的最大好处是可移植性。在目前的标准中，关于中断驱动的接收、远程执行、主动消息等需要操作系统支持的一些操作以及直接的线程支持、任务管理、动态调试设施等还没有定义。因为 MPI 主要是为了可移植性，因此它的通信效率同其它的一些系统比起来要低一些。

不同的消息传递可移植平台有不同的特点和各自的长处，没有一种确定的标准能够判断其优劣，所有的系统都力图发掘构成并行机系统的所有节点机的最大性能，如 PVM 开发出了硬件能力的80～90%。在通信性能上，大多数消息传递系统在点到点通信速度上也比较接近。目前的 PVM 系统大都分是在运输层上实现的(基于 TCP/IP 的 TCP/UDP)，现在有人研究在数据链路层上实现 PVM[6]。提供低时延的报文传递系统是一个很重要的研究课题。

3 如何解决面临的问题

从以上的分析可知，NOW 环境中存在通信时延的主要原因在于：(1)慢的网络接口访问速度。实验结果表明时延的很大一部分是用于通过 I/O 总线访问网络接口，而在局域网中，信号在物理链路上的传输时间很小。(2)复杂的网络协议开销。已有网络协议的开销对于并行计算来说太大。(3)网络结点上的操作系统不提供全局命名、全局进程调度或地址翻译。因此，应着重从以下几个方面来改善通信时延。

3.1 主动消息

主动消息(Active Message)是 Eicken 和 Culler 等提出的一种提高互连网络通信性能的异构的异步通信方式[7]。这种通信的基本思想是一个报文头部的控制信息中包含一个用户级指令序列地址，该指

令序列(即报文接收方中断处理子程序)的功能是从网络中获取报文并将它集成到计算中去。其主要特征是,消息传送起动开销低,计算与通信重迭进行,通信操作向硬件层实现的描述接近等。主动消息不同于远程过程调用之处在于它的报文处理子程序的功能不是对数据进行计算而是仅从网络中获取报文并将它集成到即将进行的计算中去,完成这个只需做少量的工作。同一般的发送/接收的报文驱动的通信方法相比,它不需要缓冲。

自从主动消息提出后,人们在不同的平台上加以实现。如 L. W. Tucker 等人在大规模并行处理机 CM-5上实现了主动消息层[8],实验结果表明主动消息是一种减小通信时延而又不限制系统灵活性的特别有用方法。

在 ATM 局域网上实现主动消息是关于主动消息的一个热门研究课题。T. von Eicken 等人在 ATM 局域网中(工作站采用 SparcStation)实现了主动消息层并把它同 CM-5机上实现的主动消息层作了比较[3]。在 ATM 局域网上获得的性能(来回时延52μs)与 CM-5上获得的性能(来回时延12μs)之间的差距主要在于 ATM 局域网中慢的网络接口访问速度。

3.2 ATM 网络的应用

利用 ATM 局域网作并行计算的主要工作是在 ATM 局域网上实现已有的基于报文交换的并行应用编程环境或通信层,如 PVM, Active Message 等,以及直接利用 ATM 的应用编程接口进行通信[3-6]。

在以 ATM 局域网作为网络平台的 NOW 上作并行计算是一个很重要的方向,这方面的工作主要有:

·如何更有效地利用 ATM 交换机的基于交换的特点使群通信尽量并行执行以提高通信效率。

·考虑如何解决 ATM 局域网中进行大数据量通信问题。因为 ATM 传输单元的大小为53字节,对于大数据量的传输分段和重组要带来比较大的时延,所以有必要考虑这一点。有些新一代的 ATM 网卡用硬件来实现这一点。

·在 ATM 的应用编程接口 API 上实现已有的消息传递系统,如 PVM, MPI, Active Message 等。

3.3 虚拟内存映象的网络接口

流线型网络接口驱动程序的设计。实验表明,网络接口驱动程序是降低时延的关键,应尽量减少操作系统的干预和数据拷贝次数,这主要通过用户地址空间与网络驱动程序内存之间的映射来实现用户级地址访问,大数据量时利用 DMA 来实现数据移动。还可以采取预分配大的缓冲区和采取先进先出(FIFO)的处理策略来减小数据拷贝次数和缓冲区管理。

3.4 多层次低时延的并行计算环境

提供多层次的并行计算环境。程序员可以根据应用的需要选择适当的层次接口。层次低的通信效率高,但程序员不得不完成一些额外的工作;层次高的,编程方便,但以效率为代价。Lewis W. Tucker 等人[6]在 CM-5机上实现的报文传送系统 CMMD,以主动消息原语作为基础,构造了多层次、低时延通信接口是一个值得借鉴的例子。

3.5 从系统的角度解决通信时延问题

线程调度机制。为了提高并行计算系统的效率,必须尽量使计算与通信重叠。要做到这一点必须与进程调度紧密结合起来,但是进程调度的开销对并行计算而言太大,而线程调度的开销相对来说就小一些。因此,须研究线程调度机制。报文传递接口标准 MPI 对线程也有所考虑。

为了满足并行计算的高速、低时延通信的需要,需要对 NOW 环境中已有的系统软件作些修改或者增加一些附加的硬件。但是,对已有的系统软件或系统结构的修改必然导致系统的通用性、可移植性等方面的损失。如何在这两者之间找到一个合理的结合点是一个很重要的研究课题。

通信是并行计算的基础,要提供有效的通信支持必须将通信问题与进程调度、全局命名等操作系统的其它方面以及报文传递协议等联合起来考虑。Berkeley 的 NOW 工程中的 GLUnix 是一个很好的尝试[1]。

参考文献

[1]Tom Anderson et al., A Case for Network of Workstations(NOW), Hot Interconnects Ⅱ, Aug. 11-13, 1994

[2]Chengchang Huang, Philip K. Mckinley, Communication Issues in Parallel Computing across ATM Networks, IEEE Parallel & Distributed Technology, 1995

[3]T. von Eicken et al., Low Latency Communication over ATM Networks Using Active Messages, IEEE Micro, 15(1)1995

[4]Sheue-Ling Chang et al., Enhanced PVM Communications over a High-Speed LAN, same to [2]

[5]M. Lin et al., Distributed Network Computing over Local ATM Networks, Special Issue on ATM LANs, IEEE J. Selected Areas in Communications, May 1995

(下转第19页)