集群高效通信机制分析*

李　涛, 王　华, 刘培峰, 刘光武, 杨愚鲁
(南开大学 计算机科学与技术系, 天津 300071)

摘　要: 集群是当今高性能计算领域的重要发展方向,随着高速互联网络硬件的发展,低效的通信方式成为制约集群整体性能的关键因素。高效的通信机制能够更好地利用互联网络硬件为高性能计算提供更高性能的通信支持,通信性能通常都远高于传统的 IP 协议。

关键词: 集群; 高性能计算; 通信机制
中图法分类号: TP393.04　　文献标识码: A　　文章编号: 1001-3695(2005)10-0257-04

Analysis on High Performance Communication Mechanisms of Cluster System

LI Tao, WANG Hua, LIU Pei-feng, LIU Guang-wu, YANG Yu-lu
(Dept. of Computer Science & Technology, Nankai University, Tianjin 300071, China)

Abstract: In the high performance computing area, cluster has been playing an increasingly important role. With the development of high-speed interconnection network, the traditional communication method becomes the main obstacle to the performanceof cluster system. This paper discusses several high performance communication mechanisms, which efficiently support high performance computing. Their performances are far better than that of traditional IP protocol.

Key words: Cluster; High Performance Computing; Communication Mechanism

在很多科学计算和工程应用领域中,对计算能力的需求越来越高,这在很长一段时间内将依靠并行处理技术来解决。集群(Cluster)是由网络连接在一起的很多工作站或 PC 机构成的并行/分布式处理系统,像一个单独集成的计算资源一样工作,它充分利用网络和单机性能的优势。集群具有投资小,研制周期短,性能价格比高,可扩展性高,使用方便等特点,目前已经成为高性能计算领域重要的发展方向[1~3]。互联网络硬件性能的提高直接提高了硬件层通信性能,然而低效的通信方式不能充分利用物理网络的通信性能,已经成为集群通信性能提高的障碍,提高集群通信性能已经成为高性能集群计算领域的研究热点。

1　通信机制的优化技术

当今应用需求对通信性能提出了很高的要求,随着计算机软/硬件技术的发展,处理器和 I/O 的性能有了大幅度提高,原始的网络通信带宽已经以指数的速度在增长,并且可靠性也有很大提高。因此,影响集群各节点间通信能力的因素由硬件转向软件,特别是通信软件,包括协议结构和算法等。通信软件的开销通常主宰着集群的通信时间,主要开销来自以下几个方面: ①软件需要穿过多个协议层; ②消息通信通常需要多个内存拷贝过程; ③信息传输过程,通信软件需要多次通过保护边界。另外还受到包括系统提供的通信服务和所支持的用户环境的影响。所以,精简通信协议、减少系统调用和数据在内存间拷贝产生的开销是提高通信带宽、降低通信延迟的重要手段,从而减小原始的网络带宽和用户的可用带宽之间的差距。

收稿日期: 2004-09-26; 修返日期: 2004-11-22
基金项目: 天津市自然科学基金项目(043600111)

目前,对集群通信协议的研究主要集中于用户级通信协议[2],其基本思想是: 网络接口卡的所有缓存和寄存器都从内核空间重映射到用户内存空间,用户进程不用跨越用户 - 内核边界就可以驱动设备,它允许通信程序直接访问网络接口硬件,消除了操作系统的参与,尽可能地将网络硬件的高性能反映到用户层。网络接口卡在发送和接收数据时可以直接访问内核级缓存,用户程序也可以直接访问它们,避免了消息的暂时缓存,提高了通信效率。

用户级的轻量级方式是通过在通信路径中减少操作系统的介入来提高性能,以获得应用程序和通信设备直接更亲密的结合,保证了在没有任何系统调用的情况下对网络接口通信缓存的访问,在用户级程序库实现。这种方式对操作系统内核的修改可能从简单的添加定制设备驱动程序,到复杂的调度器级别的深入干涉,这都依赖于需要保护程度以及允许对通信设备进行多用户访问的程度。

通信状态流水化技术也是一种重要的方法,即当主机 - 网络接口卡、DMA 或者程控 I/O 传输仍在进行的时候,一些网络接口卡可以在物理介质上被编程以流水化的方式传输数据,其性能提高在延迟和吞吐量方面都非常显著。另外,并行多个网络也是提高合计通信带宽以及减少拥塞最直接的方法,但不会减少延迟。

2　几种高效的通信机制

2.1　用户空间(User Space)协议

在 IBM SP2[4] 系统中,节点通过微通道 I/O 总线连接到高性能交换(HPS)网络上,HPS 支持科学计算和商业应用、系统

服务和 I/O 操作的消息传递通信等,多个任务可利用基于 IP 的协议来共享 HPS,但一个节点只允许一个任务使用用户空间(US)协议,因此网络适配器必须在使用 US 协议的单个进程和使用 IP 的多个进程之间共享。这样,对不需要高性能通信的任务使用基于 IP 的协议可获得更好的整个系统使用效率。

IBM SP2 系统通信软件的结构如图 1 所示,它支持基于 IP 的内核空间协议和 US 协议。在内核空间协议中,网络接口驱动(NID)将 IP 与通信适配器连接。基础通信库(BCL)在此 US 协议中由消息层(ML)和管道层(PL)构成,消息层由简单的非阻塞点对点通信库函数组成,管道层使用一对管道提供任意发送和接收进程间的可靠的、具有流量控制的、按序传送的字节流。它们具有独立的虚拟交换接口(VSI),VSI 由节点系统内存中的两个队列和适配器内存中的一些控制寄存器组成,用来提供输入/输出数据的系统 DMA 缓冲区。

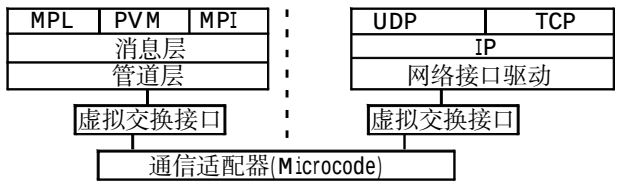

图 1 用户空间协议

节点处理器执行计算代码,消息层和管道层在协作程序中,这些代码都在同一用户进程空间中,因而没有系统调用或者内核穿越;所有的 DMA 操作由适配器启动和执行,所有的数据传输都由 DMA 操作。只有控制和状态信息在节点处理器和通信适配器通过之间可编程 I/O 来传送。

US 协议提供保证、按序、可靠和具有流量控制的高性能消息通信。适配器提供将逻辑进程标志翻译为物理地址的功能,保证了通信在已分配的区域内进行。使用令牌环协议进行流量控制,并利用超时和重传机制的确认包来保证消息的可靠传输。

2.2 活动消息(Active Message)

活动消息(AM)[5] 是一个实现低开销通信的异步通信机制,不依赖于任何硬件或软件平台,其目标是提供给用户底层通信硬件原始的通信能力,减少通信开销对应用程序性能的影响。AM 库提供编程接口给应用程序,虚拟网络段驱动程序将网络接口和通信资源抽象化,具有在网络接口卡嵌入处理器上执行协议处理的固件以及处理器和互连硬件实现。

在 AM 系统中,每条消息由消息体和指向处理该消息的消息句柄的指针组成,消息句柄一般都是基于发送方的。AM 的基本思想是使用消息头中的控制信息作为消息句柄的指针,当消息头到达目的节点时,激活与每个消息相关的句柄函数。消息句柄能够从网络中提取数据并将其合并到正在进行的计算中,而不执行数据计算,实现了通信和计算的重叠操作,而且接收方进程也减弱了消息管理功能。AM 是个单边通信范例,即不管发送方何时发送一个消息,不管接收方进程当前正在进行什么动作,都要交换消息,在另一边不需要接收操作。AM 的特殊语义可以消除消息在通信路径上对大量暂存的需要,显著地加速了通信。

AM 系统完全在用户空间中实现,消息句柄通常为用户级子例程,消除了上下文切换和穿越保护边界的开销,不需要处理系统调用,应用程序可直接访问网络接口硬件。AM 是在大规模并行系统上创建高性能通信协议、运行时环境和消息通信库的强有力范例。

随着高性能互联网络的发展,AM 在以下四个方面作了改进[6]:①修改了 AM 命名和保护模型,能够在其上创建更高级的和针对具体系统的模型;②开发了通信错误和故障模型以支持耐故障和高可用性应用;③将 AM 操作、通信事件和线程集成到简单的模型中;④使用有限的物理网络资源支持大量用户的受保护的多程序设计。现在,GAM-Ⅱ不只支持并行计算,还支持网络计算和分布式计算等。

2.3 快速消息(Fast Message)

快速消息(FM)[7] 是基于 AM 的高性能基础通信库,主要目标是使用一个简单接口将网络硬件的性能提供给应用层,在集群上提供低延迟和高带宽通信,尤其是对于短消息,FM 适用于编译器、语言运行时、通信库和某些应用程序的实现。

高性能消息传递层设计的关键问题包括主机与网络协处理器之间的分工,I/O 总线管理以及缓冲区管理等。FM 消息传递层设计中关键是 LANai 控制程序(LCP)的结构、缓冲区管理、SBus 两端两个程序的协调和主机程序的设计。FM 具有良好的编程接口,每个消息带有一个指向由发送者指定的函数指针,它不需要主机处理器参与从网络上存取消息,也不需要通过轮询来防止网络拥塞。FM 只包含很少的功能函数,包括发送长短消息和从网络上选取消息的功能,但它假设每个点上至多只有一个进程使用 FM 进行通信。FM 提供的服务保证并控制了 FM 为其内置的软件提供的内存层次,在控制了通信任务调度的同时,保证了可靠性和所需包的发送。如图 2 所示,FM 通信协议使用四个 FIFO 对,Myrinet 网络适配器内存含一个发送队列和一个接收队列,节点内核中有在非交换 DMA 区域包含一个很大的接收队列,用户空间的内存驻留了一个用于流量控制的拒绝队列。它是用户级的通信层,单一路径和 FIFO 保证了按序传送;Myrinet 网络使用虫蚀路由,利用 Credit 机制给每个发送进程分配一部分接收队列空间,并控制目的地的接收队列不满,从而保证了不丢包的具有流量控制的可靠传输;优化的流量控制以最小的开销使发送速度与接收速度相匹配。由于网络硬件可靠性高,FM 忽略错误处理。

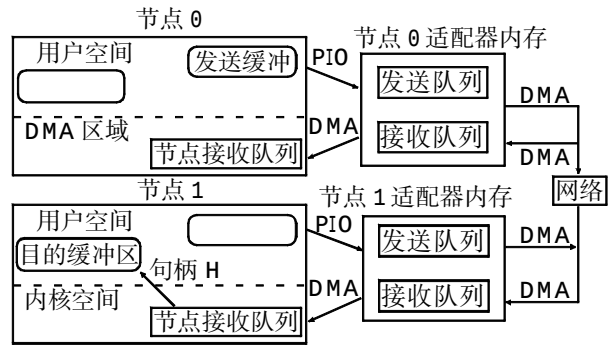

图 2 FM 通信协议

2.4 虚拟内存映射通信(VMMC)

VMMC[8] 是一种在发送者和接收者的虚拟地址空间之间进行直接数据传输的通信模型。它消除了通信中操作系统的参与,提供了用户级通信的保护,支持用户级的缓冲区管理和真正的零拷贝协议,并能使软件通信开销达到最小化,具有很高的通信性能。VMMC 可用于一个处理器上不同进程间的通信,或者一个共享内存多处理器系统中的不同处理器间的通信,或者一个局域网上不同节点上的进程间的通信。但所有这些都需要修改操作系统源代码来处理新的内存管理需求。

VMMC 通信协议如图 3 所示,每个节点上驻留 VMMC 守

护进程, VMMC 设备驱动和实现 VMMC 和 Myrinet 网络拓扑的映射的适配器控制程序。VMMC 在输入/输出关系建立阶段设置页表和保护机制, 适配器保存页表负责地址转换工作。接收进程通过输出(Export) 接收缓冲区的地址空间区域表示允许进行通信操作, 发送进程引入(Import) 它并用作传输数据的目的区域的远程缓冲区。在成功引入之后, 发送者将其虚拟内存中的数据传送到引入的接收缓冲区, 并保证数据不在目的接收缓冲区外覆盖接收者地址。

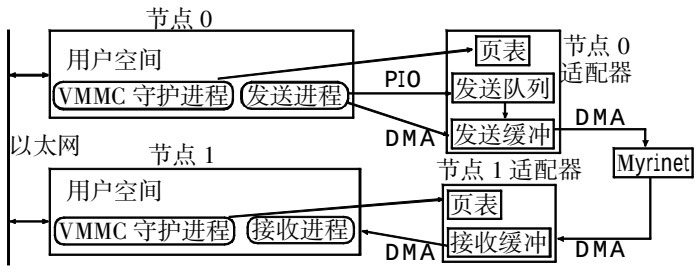

图 3　VMMC 通信协议

VMMC 支持显式更新和自动更新两种数据传输模式, 当由任何一种传输模式发出的消息到达目的端时, 消息直接被传送到接收进程的内存中, 而无须中断接收者的 CPU, 它没有显式的接收操作。发送数据时, 只有一个自动更新的本地写操作或者一些用于显式更新的用户级指令, CPU 开销非常小。

2.5　U-Net

在很多应用程序中, 短消息变得越来越重要, 对于局部通信区域中的短消息, 处理开销是决定延时的主要因素, 实现低延时通信的方法主要是通过减少每个消息的开销来实现。U-Net[9] 用户级网络接口体系结构致力于减少发送和接收消息的处理开销以及提供对网络最底层的灵活访问, 在局部区域设置中提供低延时通信, 对短消息开发出网络的最大带宽, 以及为新型通信协议的应用提供便利。

在 U-Net 中, 操作系统与硬件机制的结合可以提供给每个进程自己拥有的网络接口。依赖于实际硬件的复杂性, 一个进程所操作的 U-Net 组件可能对应一个网络接口上的实际硬件, 也可能对应由操作系统指定的内存区域, 或者对应于它们的组合。U-Net 仅用于在访问网络的所有进程间复用实际的网络接口, 加强保护边界限制以及资源消耗的限制。特别地, 像缓冲器这样的进程对每个消息的内容以及发送和接收资源的管理都能得到控制。

U-Net 给予用户进程对网络设备的直接受保护访问, 在用户级编程库中实现, 它将非常低级的机制暴露在用户级。互连结构被虚拟化为一系列端点, 一个端点就是一个内核内存缓存, 再加上为主机适配器同步而设置的一个发送队列和一个接收队列。操作系统的功能被限制于将一个和更多这样的端点通过特定的系统调用重映射到用户进程的内存空间, 此后的操作不需要更多操作系统内核的干预。如果用户进程需要的端点数超过了网络接口卡直接支持的可用端点数目, 必须由操作系统模拟额外的端点, 提供相同的功能, 但是性能有所降低。U-Net 提供了良好的通信性能, 但通信是不可靠的。

2.6　并行基本接口(BIP)

BIP[10] 是在 Myrinet 上实现的用于消息传递并行计算的网络通信接口, 通过直接被应用程序访问的库接口, 尽量减少对系统内核的访问, 提供一个具有底层功能的高速通信协议。

BIP 提供多种功能来得到参数和结构常量, 发送和接收的阻塞和非阻塞的通信原语, 以及一个根据集合通信模式实现的发送 - 接收范例。BIP 通信有两种不同的语法, 长消息的发送和接收具有会合语义, 接收操作需要在发送操作之前发出; 短消息需要存放在目的端的循环队列中, 即使没有相应的接收操作, 发送操作也不会被阻塞。除了很短的消息以外, 都使用缓存模式来管理。BIP 消息可以加上标志进行区分, 发送操作的其他参数是数据和目的地的逻辑编码, 接收操作不指定特定的发送者, 但会检查标志, 其他参数是接收消息的缓冲区和可以接收的最大长度。

在初始化阶段, BIP 建立一个表并给每对处理器选择(或允许用户选择) 一个静态路由。在 Myrinet 上, BIP 消息根据发送者提供的路由信息可以通过多个交换机进行路由。BIP 通过透明的将消息分段分组而使通信路径流水化, 分组的大小依赖于总的消息大小, 目的是在通信的每一端都允许 Myrinet 适配器的不同 DMA 引擎以流水线的方式工作, 并达到最好的负载平衡。另外, BIP 还通过消除网络接口卡的受保护复用而获得了最优性能: Myrinet 适配器的寄存器和内存区域全部暴露于用户级访问。

发送和接收在非阻塞语义中也是可用的, 可以检测或者等待异步发送和接收调用的完成。由于发送和接收操作是完全独立的, 因此可以任何方式将它们混合使用。非阻塞原语允许计算和通信操作在适当的时候重叠, 但是对每个挂起的标志, 不能有超过一个发送操作和一个接收操作。

BIP 在消息传输中加入序列号并进行 CRC 校验, 可以检测介质上的任何错误或丢包, 但不实现任何恢复策略, 依赖 Myrinet 良好的网络硬件, BIP 保证消息的可靠传送。

3　高效通信机制的比较与分析

集群中最重要的通信性能参数为延迟和带宽, 延迟和带宽的测量受测量所用的硬件平台的影响很大, 如 CPU 和内存速度、I/O 系统结构、网络接口卡和传输链路等, 有可能使测量的数据不能相互比较。特别地, CPU 的不同速度会导致所测量的延迟具有较大的差异, 测量技术的不同会导致所测量的带宽具有较大的差异。表 1 给出了 TCP/IP 以及本文讨论的几种高效通信机制的延迟和带宽在不同平台上的测量结果。

表 1　消息传递系统的点到点通信的延迟和带宽比较

Mechanisms	Latency(μs)	Bandwidth(Mbps)
TCP/IP (Ethernet)	113.8	6.6
US (HPS)	44.8	34.9
AM (FDDI)	14.5	12
FM (Myrinet)	11	77
VMMC (MC)	9.8	108
U-Net (FE)	30	12.1
BIP (Myrinet)	4.3	126

可以看出, 由于互联网络硬件性能的提高, 以及采用简化通信协议、避免操作系统调用和消息的暂时缓存以及通信状态流水化的使用, 各种高效通信机制的延迟和带宽远远要好于 TCP/IP 的性能, 为上层计算环境提供了高性能通信。

US 使用避免系统调用的优化方法, 提供保证、按序、可靠, 且具有流量控制的高性能通信, 它使用超时重传进行错误处理, 其缺点是同一时间只能有一个进程使用 US 协议。U-Net

使用简化通信协议的方式,尽量减少发送/接收处理开销,而且对短消息的支持比较好,但当用户进程端点数超过网络接口卡上接口数时性能下降情况比较严重。这样,就限制了其性能的进一步提高。AM 是完全实现在用户级的低开销的异步通信机制,FM 是基于 AM 的高性能基础通信库,高带宽、低延时,尤其是对短消息的支持比较好,它忽略错误处理,直接交给下层硬件实现,表 1 中所示性能差异主要来自于硬件环境的不同,尤其是带宽相差很大。VMMC 使用避免消息暂存的优化方法,能够实现真正的零拷贝,但有的需要修改操作系统源代码。BIP 使用通信状态的流水化方式进行优化,给用户暴露了很好的底层硬件性能,使用 CRC 校验,但不进行错误恢复,而且它依赖于硬件的流量控制功能。它们实现了非常低的延时和非常高的带宽,尤其通信协议简化和流水化的使用,更好地利用了互联网络的原始带宽,进一步提高通信性能。

4 总结

在集群不出现故障的情况下,主要的性能瓶颈不是计算资源,而是怎样提供低延迟、高带宽的连接和怎样以有效的低层通信协议来提供高层 API,降低消息传递的延迟。本文讨论的几种高效通信机制使用了简化通信协议、避免操作系统调用和消息的暂时缓存,将内核级缓存重映射到用户内存空间,完全在用户级实现通信系统,尽可能地将网络硬件的高性能提供给用户层,通信流水化的使用进一步提高了通信性能。另外,还可以并行使用多个通信网络或者其他一些更加低级的通信机制。高效通信机制的使用获得了高性能通信,充分利用了互联网络的原始带宽,更好地发挥了集群的整体性能。

参考文献:

[1] ttp://www.top500.org/ [EB/OL].

[2] 郑纬民,石威,汪东升,等.高性能集群计算:结构与系统[M].北京:电子工业出版社,2001.

[3] K Hwang, Z Xu. Scalablde Parallel Computing Technology, Architecture, Programming [M]. Beijing: China Machine Press, 1999.

[4] M Snir, P Hochschild, D D Frye, *et al*. The Communication Software and Parallel Environment of the IBM SP2 [J]. IBM System Journal, 1995, 34 (2): 205-221.

[5] T V Eicken, *et al*. Active Messages: A Mechanism for Integrated Communication and Computation[C]. Proc. of the 19th Annual International Symposium on Computer Architectures, 1992. 256-266.

[6] A Mainwaring, D E Culler. Generic Active Message Applications Programming Interface and Communication Subsystems Organization [R]. Division of Computer Science, University of California at Berkeley, 1996.

[7] S Pakin, M Lauria, A Chien. High Performance Messaging on Workstations: Illinois Fast Messages (FM) for Myrinet [C]. Supercomputing '95, San Diego, CA, 1995.

[8] C Dubnicki, L Iftode, E W Felten, *et al*. Software Support for Virtual Memory-Mapped Communication [C]. Proceedings of the 10th International Parallel Processing Symposium, 1996. 372-381.

[9] T V Eicken, A Basu, *et al*. U-Net: A User-level Network Interface for Parallel and Distributed Computing [C]. Proceedings of the 15th Annual Symposium on Operating System Principles, 1995. 40-53.

[10] L Prylli, B Tourancheau. BIP: A New Protocol Designed for High Performance Networking on Myrinet [C]. The PC-NOW Workshop, IPPS/SPDP, 1998.

作者简介:

李涛(1977-),男,山东泰安人,讲师,博士研究生,主要研究方向为并行与分布计算、高性能网络;王华(1979-),男,硕士研究生,主要研究方向为集群计算;刘培峰(1981-),男,硕士研究生,主要研究方向为并行机系统结构;刘光武(1969-),男,硕士研究生,主要研究方向为通信系统;杨愚鲁(1961-),男,博士生导师,主要研究方向为并行机系统结构、网格计算、可重构系统。

(上接第 256 页)

(4) Local 用来指明隧道封装的源 IP 地址,在该方案中应为 HA 的 IP 地址。

(5) ttl 用来设置隧道的生存期。

删除隧道的命令:Iptunnel Del[tunnel name]。其中 Del 表明要删除一条隧道,Tunnel Name 为要删除的隧道名。

另外,由于隧道在这里作为一种虚拟的网络接口,因此需要用 Ifconfig 命令进行驱动接口的配置。

在该方案的实现中,HA 除了要按上面的命令建立隧道以外,还必须在应用进程中指定隧道设备发送数据。要实现指定设备发送,可以通过 setsockopt 函数设置发送设备。

5 结束语

本文提出了一种基于 Linux 下的 Netfilter 技术实现移动 IP 中家乡代理多重绑定机制的方案,对该方案的设计思想,设计框架以及具体设计方案作了详细的介绍,并且给出了该方案实现中关键技术的解决方法。该方案利用 Netfilter 框架实现对数据报的过滤处理功能,在 HA 上将发往具有多重绑定的 MN 的数据截获到用户空间,由 HA 的用户态进程对这些数据进行处理和发送。该方案避免了在内核态进行数据的复制,因此不会对内核运行的稳定性造成任何影响。另外该方案在设计上具有框架清晰等特点,在实现上巧妙地利用了 Netfilter 的用户配置工具 Iptables 提供的配置功能,使得方案的实现灵活简便。

参考文献:

[1] Perkins. RFC 3344 IP Mobility Support for IPv4[S]. Aug. 2002.

[2] Hiroshi Esaki. Multi-Homing and Multi-Path Architecture Using Mobile IP and NEMO Framework[C]. Proceedings of the International Symposium on Applications and the Internet, 2004. 26-30.

[3] M Kassi. Mobile IPv6 for Multiple Interfaces[S]. Internet-Draft, 21, 10, 2003.

[4] Ye Min, hua, Liu Yu. The Mobile IP Handoff between Hybrid Networks[C]. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2002. 265-269.

[5] 博嘉科技. Linux 防火墙技术探密[M]. 北京:国防工业出版社,2002.

作者简介:

谭敏(1980-),女,黑龙江哈尔滨人,硕士研究生,主要研究方向为无线网络技术与应用;田霖(1980-),女,山东日照人,硕士研究生,主要研究方向为无线通信技术;夏寅贲,博士研究生,主要研究方向为下一代网络与无线通信技术。