基于机群系统的通信协议性能比较、分析与研究*)

Comparison, Analysis and Research of Communication Protocol and Performance on Cluster

魏英霞　舒继武　王鼎兴　郑纬民

(清华大学计算机科学与技术系　北京 100084)

Abstract　As we all know, communication technology is one of the key factors improving the performance of Cluster. It influences parallel speedup, parallel computing efficiency, scalability and system's application domain greatly. To provide low latency and high bandwidth is one of the main goals of research on Cluster. In this paper, the main causes of communication overhead are analyzed, and several methods to enhance communication performance are proposed. Some current communication protocols with high efficiency are listed, and their innovations and shortcomings are discussed. At last, we present the following research direction in that area.

Keywords　Cluster, Communication protocol, Zero-copy technology, Communication overhead, Collective communication

1. 引言

机群(Cluster)系统是实现并行计算的一种新主流技术。它通常由以高速 LAN 连接的工作站、微机或 SMP 组成。近几年来,新的高性能通信技术的出现,使得 Cluster 越来越流行。快速 LAN(ATM、FDDI、光纤通道和 Myrinet)无论是在通信延迟还是带宽上都可与 MPP 相比拟。然而,没有通信协议实现上的根本变化,软件开销将成为提高通信性能的瓶颈,应用程序很难从这些新技术中获益。例如在 Sun SPARC20 工作站上,640Mbps 的 Myrinet 上 TCP 的最大网络带宽只有 42.6Mbps,远远小于它的物理带宽。

在机群系统中,通信速度是影响并行计算性能的关键因素。建立低延迟、高带宽的可靠通信协议是机群系统研究的主要目标之一。传统的 TCP/IP 协议是为广域网设计的网际互连协议,它提供了复杂而强大的诸多功能,如路由选择、流量控制、差错控制、出错重发机制、拥塞机制、连接管理和缓冲管理等,从而带来了很大的通信开销,特别是由于协议层次多、数据拷贝频繁所引起的通信开销。因此,需要有新一代的有效的消息传递层,利用硬件性能的提高,向应用层提供更好的服务,以提高通信系统的性能。为了降低通信软件处理开销,许多项目都在研究高性能通信软件的设计(如:FM、AM、U-Net、VMMC-2、PM 和 BIP 等),人们用各种方法简化和改善原有的通信协议并提出新的通信协议,以提高机群系统的整体通信性能。本文主要综述和比较了目前常用的几种高效的通信协议,介绍了它们的实现方式、采用的技术以及它们的优点和局限性,并指出了今后机群通信系统的研究方向。

2. 通信开销的主要来源

并行应用程序中一次点对点的通信通常要经过这样一个过程:首先应用程序调用通信函数将要传送的数据拷贝到协议缓冲区,协议则将数据分拆成适合链路传输的数据包,再通过网络把数据包传送到目的节点;然后经过相反的过程,最终将数据输入接收进程。机群系统中通信开销主要有如下四部分:

①缓冲管理:数据缓冲用于消息传输时包的组装和分拆以及出错重传。如 Solaris 采用 mbuf 方法[13]静态地分配缓冲区,因分配的空间较小,容易发生缓冲区满的现象,增加了阻塞延迟,从而增大了通信开销。

②数据拷贝:数据被发送时要经过从用户缓冲区到 Socket 层缓冲区的拷贝,协议层进行协议处理时的拷贝,再到接口缓冲队列的拷贝,最后,由驱动程序通过 I/O 总线拷贝到网络适配器上发送。在 Sun SPARC20 工作站上,内存带宽只有 70MB/s,即拷贝一个以太网包(1400 字节)需 30 微秒。多次的数据拷贝造成了很大的通信开销。

③协议处理时间:其成因主要有复杂的协议机制与校验和计算。传统网络协议是基于低带宽、高差错率的物理链路,为保证可靠数据传输而设计的,其协议机制很复杂。校验和计算非常费时[14],在 Sun SPARC20 上计算 1k 字节数据的冗余校验和要 60 微秒。

④操作系统开销:TCP/IP 是在操作系统核心实现的,操作系统的系统调用和原语为网络协议实现提供了底层的软件支持,但由此引起的上下文切换、页面调入/调出、I/O 设备启动、中断响应等操作系统处理也带来了不容忽视的开销。比如,在 Sun 3/60 系统上 TCP/IP 对一个数据包的协议处理时间是 100 微秒,而操作系统开销则高达 240 微秒。

3. 通信性能的改善方法

通过上述对通信开销的分析,我们可以从以下几个方面实现对通信软件性能的提高。

3.1　精简通信协议

通信开销很大程度上是由协议层次多、数据拷贝频繁引起的。另外,为满足各种用户的需求,通用的网络接口和协议增加了许多其它的功能,这些功能也带来了额外的开销。相比之下,机群系统的网络系统分布范围小,通信链路可靠性高,

*)本文研究得到国家自然科学基金(No. 69933020)资助。魏英霞　硕士生,主要研究领域为并行处理。舒继武　博士,主要研究领域为大规模科学与工程计算中并行算法、并行处理技术及并行应用软件的研究。王鼎兴　教授,博士生导师,主要研究领域为并行计算机体系结构、并行计算等。郑纬民　教授,博士生导师,主要研究领域为并行计算机体系结构、并行计算等。

系统结构相对简单，使用一个提供有序、可靠传输的简单底层通信层便足够了，因此对通信协议进行精简可以减少不必要的通信开销。

所谓的精简，包括两部分：一是功能的精简，就是删除不必要的和冗余的功能；二是协议层次的精简，即合并各层的功能，使通信协议变为一层，以达到减少数据拷贝的目的。比如，在操作系统 Solaris2.4 中，通信协议由网络驱动程序、数据链路层(DLPI)、IP 层、TCP 层和 Socket 接口组成，由于数据链路层 DLPI 已经提供了不保证数据包无差错传送的基本数据通信功能，因此可以在它的基础上实现一个保证数据可靠传送的模块，以取代复杂的 TCP/IP 协议和 Socket 接口，这样新的通信协议不论在结构上还是在功能上，都比原有的协议要简单得多。

3.2 减少多余的数据拷贝

在通信软件中，大部分的开销来自在用户空间和内核空间之间或通信软件的不同层次之间拷贝消息。因此，探索如何减少这些拷贝是很有意义的。基本通信层(BCL)按其数据拷贝的次数可分为双拷贝、单拷贝和零拷贝[15]。其中用户空间协议为双拷贝，例如 IBM 的 SP；FM(Fast Message)为单拷贝；而 Princeton 的 SHRIMP 则采用了零拷贝技术。零拷贝技术是减少数据拷贝次数、实现高性能通信机制的重要方法之一。所谓零拷贝技术，就是在消息的传递过程中，不需要任何额外的内存拷贝，消息数据从发送方用户缓冲区通过网络直接传送到接收方用户缓冲区。

实现零拷贝有两种策略：一种是通信软件在核心空间开辟一块符合 DMA 要求的缓冲区，映射到用户空间，让用户去管理，并要求用户把传递的数据放在该缓冲区里[9]。这样做的缺点是用户必须承担复杂的缓冲区管理工作，在很大程度上限制了用户编程时对内存的使用，而且分配多大的 DMA 缓冲区也不好控制。另一种方法是利用操作系统核心在通信前或通信进行中将用户的数据缓冲区转换为 DMA 缓冲区[10]。这种策略对用户传递的数据的放置没有任何限制，即用户可以在任意地址空间为要传送的数据分配内存，但要求用户在发送之前调用类似 flush 的操作，确保数据真的写入内存；另外，还要调用类似 pin 的操作，确保相应的内存页不被换出。

3.3 用户态通信协议

减少操作系统的额外开销的一个重要的方法是在用户空间实现一个用户态通信协议层，使得此协议能够旁路操作系统的影响，直接对网络硬件设备进行操作。这样，在应用层实现通信协议就可以减少操作系统核心的干预和相应的开销，提高通信效率[10]，从而避免操作系统调用的时间开销，因此用户层消息传递是减少通信延迟，提高带宽的重要手段。在机群系统中，可通过把操作系统从通信的关键路径上移去来获得高效的通信性能，这个方法用于 Hamlyn，Cranium，U-Net，SHRIMP 等系统。如在 3.1 节的精简通信协议中，通过使用内存映射机制 mmap，应用程序可以在用户空间直接与网络接口进行通信，访问网卡上的寄存器和内存，而不用操作系统的干预，这样就避开了操作系统的额外开销，同时也减少了数据拷贝的次数。

3.4 Active Messages 技术[18]

实现用户态通信协议和对通信协议进行精简这两种方法都是对传统协议的实现方法所作的改进，而 Active Messages[6]则是一种全新的通信机制，为实现更高效的通信系统性能提供了可能。

在 Active Messages 通信方式下，消息的数据结构与传统的消息传递机制有所不同，它除了包含通常的数据项外，还增加了两项内容：消息处理程序指针及其参数。当消息到达目的节点后，系统立即产生中断调用，并由中断处理机制启动消息处理程序。消息处理程序的功能是从网卡上取出消息并向发送方发送一个应答消息，然后返回原来被中断的应用程序。Active Messages 的主要特点有：

①Active Messages 的消息处理方式与网络硬件的处理方式相一致。

②Active Messages 通信机制是一种消息驱动的异步通信方式。消息驱动可以使 CPU 获得较高的利用率；异步通信可以实现通信与计算的重叠。

③Active Messages 通信机制能够简化缓冲管理。由于在用户程序中已经预先分配了存储空间，接收到的数据可以直接存放到那里，因此在接收方可以取消缓冲。

4. 当前的通信协议性能比较与分析

为了降低通信软件处理开销，国内外对通信协议的设计与实现作了大量的研究，提出了许多有效的方法和技术，以提高机群系统上的整体通信性能。下面介绍目前几种高效通信协议的实现方式、采用的技术及其优点和局限性。

4.1 基于 URTP(User-Level Reliable Transport Protocol)的高性能 MPI[1]

Jehoshua Bruck 等人实现了一个基于机群系统的高性能 MPI。该系统的一个进程逻辑上由四个软件层次组成(见图 1)。该系统提供了对 URTP 层和 MPI-CCL(Collective Communication Library)层的高效率的设计和实现。

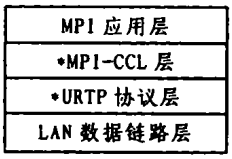

图 1 系统的四个软件层

URTP 协议层利用 LAN 层提供的多点播送和广播能力，向上层提供了可靠的传输协议，具有点到点和多点播送能力；MPI-CCL 层把所有的 MPI-CCL 例程映射到 URTP 支持的接口，还负责用户消息的分包和调度及发送端缓冲区管理，由于 LAN 数据链路层的广播是不可靠的，发送端需要为每个发出的包留一个拷贝，直到目标集中的所有处理器都确认接收。

URTP 的实现由内核扩展和用户层库两部分组成。内核扩展可以不经过用户空间而处理通信开销，快速地丢弃多点播送包，同时减少两个进程之间的通信开销，较大程度上提高了系统的性能。目前的 LAN 由于介质错误而引起的包丢失率很小，大多数包的丢失都是由接收端缓冲区满造成的。URTP 把包从内核缓冲区移至用户层缓冲区，尽可能快地释放内核缓冲区，减少了因接收缓冲区(内核缓冲区)满而带来的包的丢失。URTP 的用户层库通过使用修改过的滑动窗口协议实现可靠的传输。

现有的并行编程环境，如 PVM、EXPRESS 和 IBM's MPL(Message Passing Library)，都是建立在 LAN 之上的，它们的全局通信是建立在点到点通信之上的，其性能很差。该

方法充分利用了低层 LAN 的广播特性，简化了全局通信的实现，提高了系统的整体性能。其不足是：URTP 的性能不够好，主要是对长消息的延迟较大；另外，URTP 缺少用户发送缓冲区的管理、分包(packetization)和聚合(reassembling)等功能，使其不能成为一个独立的工具；还有，它不支持多个 MPI 应用同时运行。

4.2 UIUC 的 FM[2]

Illinois FM 是 UIUC 开发的一个低层通信层，它提供可靠传输和有序传输等关键保证，以及对通信工作的调度控制，通过提供缓冲把处理器从网络上去耦。所谓去耦，即当处理器忙于计算时，网络可以继续发送数据，这些数据被放入缓冲区，当处理器空闲的时候再从缓冲区中取数据，这样发送者不必等待接收过程完成，即不必等到接收端从网络上析取消息之后才返回。

FM 完全在用户空间实现通信协议，避免了系统调用带来的开销。FM 使用固定的包格式简化了网络接口控制程序的队列管理，它允许消息传输期间发送端、网络接口和接收端的协议处理之间的重叠。当包到达一个节点时，网络接口将其以 DMA 方式移入主机的存储器(DMA 缓冲区)，这一过程保证了好的带宽、包的快速传输和对网络资源的立即使用。

FM 还支持收集-分散、层次交叉和接收端流量控制。

收集-分散 通过执行一系列 FM_send_piece( )调用，用户可以使消息由任意多的片段组成，其大小任意。同样，接收方使用一系列的 FM_receive( )调用，把消息分解成任意多的片段。每个调用组成/析取所希望的字节数，并且两端片段的个数和大小不一定要匹配。收集-分散功能允许字节流化处理消息头的接上和除去，在此之前这需要拷贝传输的数据。

层次交叉 字节流抽象的第二个好处是 FM 同接收方应用层之间的可控的交叉。层次之间的交叉可以消除存储消息的层次缓冲。例如，在 MPI-FM 中，使用 FM1.x，到来的消息由 FM 来处理，而缓冲区管理发生在 MPI-FM，由于两个层次之间无法进行直接的信息交换，数据接收由用户事先投寄一个 MPI 接收调用来实现。层次之间的交叉使得消息数据可以直接被送入应用层目的缓冲区，而不是使用分层缓冲区，这样就节省了另一个拷贝。

接收端流量控制 FM2.x 的接口还提供接收方流量控制，允许接收方控制来自网络的消息的处理速度。这可以避免缓冲区被用完和存储器拷贝，去除了消息丢弃操作，简化了缓冲区的管理，使得零拷贝传输成为可能。

FM 的关键特征(收集-分散、层次交叉、接收端流量控制)简化了上层的 MPI 的实现，把大部分低层的 FM 的性能传递给应用。它的不足之处在于：不支持多用户和多进程；缺乏与传统网络和网络协议结构的协同工作能力。

4.3 UIUC 的 MPI-FM[3]

UIUC 以 FM 为底层通信层改写了 MPI，实现了基于 FM 的高效 MPI 系统 MPI-FM。

MPI-FM 建立于 Myrinet 网络之上，它是以低延迟和高带宽通信为目标重新设计的，所有的决定，包括接口中包含哪些服务，都是基于对性能的考虑而作出的。MPI-FM 对 FM 做的改变有：增加了一个简单的收集(gather)原语，支持像增加或除去消息头这样的普通操作，还加入了一个向上调用(一个定义于上层而由下层调用的函数)，使得在层次之间可以交换目标缓冲地址等关键的信息。同时实现上述优化，在所有的相关处理器(发送端、接收端和网络接口)之间进行平衡，较大地提高了 FM 的性能。

MPI-FM 减少了存储器之间的拷贝，其中主要一部分来源于在接收端对长消息的合段(reassembly)操作。如图 2(中)所示，FM 中的长消息在发送端被分成若干个包，在接收端，它们被放在了一个合段缓冲区。与短消息的传送相比，长消息的传送多了一个在合段缓冲区中拷贝的步骤。为了减少这部分开销，MPI-FM 直接在目的缓冲区(图 2，右)进行包的合段。每当新的段到来的时候，FM 使用向上调用向应用代码询问使用哪个缓冲区作为合段缓冲区。在向上调用中增加了改变消息指针的功能。当函数读到用来确定消息的身份的信息(消息头)后，它把指针增加，使其指到用户数据，FM 则从更新后的位置拷贝该段。

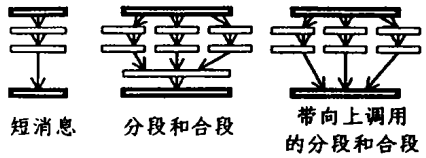

图 2 MPI_FM 中的分段和合段

MPI-FM 还去除了在发送端聚集包时的拷贝。在发送端，原有的 FM 把具有相同的消息头的消息片拷贝到一个连续的地址空间，然后再发送，从而带来了额外的开销。MPI-FM 使用一个新的原语 FM_gather24，允许一个消息由许多片组成，从而避免了多余的拷贝。

MPI-FM 对于短消息或中等长度消息的传输性能要比 SP2 和 Cray T3D 上的通信系统好，缺点是对长消息的延迟和带宽不如后两者；要想进一步提高协议的性能，需要对 API 进行一些根本上的改变。

4.4 基于 Myrinet 的 GM[17]

GM 是 Myrinet 上的一种基于消息的通信系统。它有如下特征：GM 的通信开销很低，在各种体系结构上，一个包的开销约为 1 微秒；即使网络有错时，GM 仍能提供节点间有序、可靠的传输；GM 支持由 10,000 个节点组成的机群；GM 提供两级消息优先级，允许高效、无死锁、有限存储的消息转发；只要操作系统分配足够大的 DMA 缓冲区，GM 允许用户最大传送 2^{31}-1 个字节的消息。

GM 的编程模型：GM 的模型是非连接的，用户软件同远端端口(port)进行通信之前，不需要建立连接，用户软件只需组建一个消息，然后把它发往网上的任何一个端口即可。GM 维护网络中每两个节点之间的可靠连接，在这些可靠连接之上实现端口之间的通信，以此来提供端口间可靠、有序的传输。主机、进程和端口之间的关系如图 3。

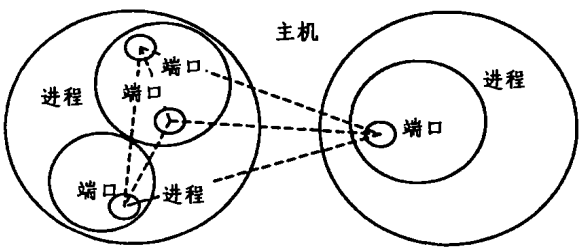

图 3 GM 的端口和进程

GM 只实现了具有同一优先级、相同发送端口和接收端口的消息的有序传输。具有不同优先级的消息不进行排序。典型的 GM 应用或者只使用一种优先级，或者使用高优先级通

道进行控制消息的传送。

GM 中的发送和接收都由隐式的令牌(tokens)来规范，令牌代表在各种内部 GM 队列中分配给用户的空间。初始化时，每个用户都隐式地拥有发送令牌和接收令牌。只有当用户拥有发送或接收令牌时，才能调用相应的函数，并释放令牌。

通过使用上述机制，GM 提供可靠的、无死锁的单跳转发。

由于 GM 是轻权通信层，它有下列不足：GM 不能向非 DMA 缓冲区发送消息，同时也不能从非 DMA 缓冲区接收消息；GM API 不直接支持 gather、scatter 等操作。可以迂过在 GM 之上建立一个重权接口来解决上述问题。

GM 是最健壮的驱动程序。对于小消息，BIP 和 FM 以及其它用户空间驱动器的延迟要比 GM 低一些，但它们的健壮性不如 GM；而对于大消息，大多数其它驱动程序的开销比 GM 要高，带宽也不如 GM。

4.5 Princeton 的 VMMC-2

最初的虚拟内存映射通信模型(VMMC)提供发送端和接收端虚拟地址空间之间的直接通信，但是它没有对高层面向连接通信 API 进行较好的支持。VMMC-2 是对基本 VMMC 的扩展，主要设计和实现了三种机制：(1)一个面向地址转换的用户控制 TLB(UTLB)机制，使得用户库可以动态地管理固定空间的大小；(2)一个传输重定向机制，避免了接收端的拷贝；(3)数据链路层的可靠通信协议，避免了发送端的拷贝。它把硬件的大部分性能传递给用户通信，同时提供高层 API 的全部功能。

UTLB 在 VMMC-2 的数据传送过程中，必须锁定(pin)用户缓冲区，数据传送结束后再解锁(unpin)。这样做的难点在于用户进程不知道物理地址，而网络接口需要物理地址来初始化 DMA。VMMC-2 通过引用一个独立的地址转换机制来解决这个问题，称之为 UTLB。UTLB 由数组组成，每个进程拥有一个数组，记录属于该进程的虚拟内存区域和被锁定在主机上的物理内存的页的地址。每个 UTLB 都是由内核空间的驱动器分配的。

与网络接口上地址转换的其它方法相比，UTLB 机制可以处理任何用户缓冲区，实现起来却没有中断驱动方法那么复杂。UTLB 的另一个重要优势是它简化了网络接口的设计。当用户发出请求时，网络接口直接从 UTLB 上获得转换信息，而中断驱动的方法则要求网络接口和主机处理器之间进行交互。

传输重定向机制自然地扩展了虚拟内存映射通信模型，支持面向连接的 API，实现了数据传输的零拷贝。传输重定向机制的基本想法是为不知道最终接收缓冲区地址的发送端提供一个缺省的、可重定向的接收缓冲区。当发送端发送数据时，它总是把数据发送到缺省的缓冲区。当数据到达接收端，重定向机制检查是否投寄了重定向地址。如果没有投寄重定向地址，数据将被移入缺省缓冲区；当接收端投寄了接收缓冲区地址时，再把数据移入接收缓冲区。

可靠通信 VMMC-2 在数据链路层实现了网络接口间的一个简单的重传协议。它的策略是缓冲发送端的包，在必要的时候重传。每个节点为系统中的所有节点维护一个重传队列，可用的缓冲区是动态管理的，所以不用为节点预留缓冲区。每个包带有唯一的顺序号(对相同的发送接收对而言)。接收方确认到来的包，发送方收到的每个确认都将确认(并释放)该顺序号之前的所有包。在接收方没有缓冲，当一个包丢失后，所有后续包都将被丢弃。

4.6 其它高效通信协议

关于通过高效通信协议来传送更好的通信性能方面还有许多其它相关研究。由于篇幅的关系，在这里只简单地列举其中的一些，不作详细介绍。

高性能通信层 AM(Active Message)[6]是最早实现的高效率通信层之一。根据网络上消息大小分布的规律(在各种网络和应用中，大多数通信量由短消息组成)，AM 提供用来传输短消息的专用低延迟原语，提高了短消息的性能，从而提高了整个通信的性能。它的不足之处是不提供有序传输。

Fast Sockets[7]是构建于 AM 之上的一个 Berkeley Sockets 实现。它去除了层次接口的不必要的拷贝。在 Fast Sockets 中避免拷贝的技术——接收投寄，与 FM 2.x 中的层次交叉相类似，都是把到来的数据直接送入目的缓冲区。缺点在于 Fast Sockets 不支持分包，因此网络可以处理的消息的大小是固定的。

Duke 大学的 Trapeze[8]项目主要研究了消息分包和流水线之间的相互作用。Trapeze 是面向 Myrinet 网络的一个通信层，它使用流水线技术来减少网络上的传输延迟。Trapeze 同 FM 用户层设计的不同在于它的优化内核中对页的传输的设计。它的缺点在于只支持固定大小的包的传送，并且消息必须来自内存中连续的地址空间；不支持端到端的流量控制、可靠和有序数据传送。

U-Net(User-Level Network Interface)[9]是一个高性能用户层通信层。它提供缓冲区管理，硬件上的多路分解，但是没有流量控制，因此可能由于溢出而产生数据丢失。与 FM 不同的是，U-Net 试图通过执行一个 DMA 传送把数据直接送入用户缓冲来避免内核存储间的数据传送。这个特征的缺点在于用户必须事先声明要用于通信的存储区域，以允许库永久地控制它们。U-Net 的一个新版本——U-Net/MM 正在开发之中，它通过在网络接口增加一个 TLB 和操作系统的虚拟存储子系统来解决上述限制。这个机制将允许网络缓冲页被动态地锁定(pinned)或解锁定(unpinned)，这样消息可以从应用的地址空间的任何一部分传送到另一部分。

PM(Parallel Message)[11]与 FM 一样，在由 Myrinet 连接的工作站机群系统上运行，它提供流量控制和缓冲区管理。与 FM 的主要不同是它具有优化了的流量控制机制和允许传输任意大小的包。不足之处是：对短消息的延迟相对于 FM 要大一些；不支持通道保护和数据的实时传送。

BIP(Basic Interface for Parallelism)[12]是另一个基于 Myrinet 开发的通信层，它有一个较传统的消息传递接口，带有阻塞和非阻塞的发送/接收原语，它是专门为支持 MPI 和 PVM 等标准的消息传递库而设计的。它的缺点在于不支持可靠传输。

小结 机群系统的一项重要功能是在各处理节点的进程之间提供高效、可靠的通信服务，通常具有性能高、可扩展性好、可用性好等特点，而通信协议正是实现上述功能的关键。本文通过对传统网络接口及通信软件中的开销的分析，指出协议处理是通信延迟中的主要成份，提出了几种改善通信协议性能的方法。阐述了一些典型的高效通信协议的设计和实现，以及它们的优点和不足之处。

系统硬件(总线结构、cache 存储器层次和 CPU 速度)所允许的有效带宽离快速以太网的理论峰值性能还很远。设备驱动程序和系统总线之间复杂的接口是影响性能的主要因素

之一，因此如何改善设备驱动程序、在协议中加入错误控制等特征是进一步研究的方向之一。

今后还可以对用户级通信软件做进一步的优化和实用化工作，使得它既能充分发挥通信硬件的性能，又能方便地用于工具软件和应用程序的编程。另外，可考虑在将来的硬件设计中，改进网络接口的功能，如：实现 FIFO 的流水线工作方式，增设通信处理器以及提供 DMA 机制等。

为了在商业计算机组成的网络基础上建立高效的服务器，今后可以进一步探索把网络协议部分或全部移至用户层的方法，以此来减少协议开销。总的框架是实现一个用户层的数据传输协议，并使其与网络接口较好地结合起来。这样可以高效地在共享虚拟内存和消息传递等模型上实现已有的高层协议，从而把硬件的大部分性能传递给用户通信，同时提供高层 API 的全部功能。共享虚拟存储模型是用户比较容易接受的编程方式，研究通信系统如何高效支持这种编程模型，解决通信瓶颈问题是非常有意义的。

机群系统的发展对通信技术提出了更高的要求，要解决上述问题，仍需我们不断的努力。

参 考 文 献

1 Bruck J，Dolev D，Ho Ching-Tien. Efficient Message Passing Interface (MPI) for Parallel Computing on Clusters of Workstation. Journal of Parallel and Distributed Computing，1997，40：19～34

2 Pakin S，Karamcheti V，Chien A A. Fast Message(FM)：Efficient，Portable Communication for Workstation Clusters and Massively-Parallel Processors. IEEE Concurrency，1997，5(2)：60～73

3 Lauria M，Chien A. MPI-FM：High Performance MPI on Workstation Cluster. Journal of Parallel and Distributed Computing，1997，40：4～18

4 Lauria M，Pakin S，Chien A. Efficient Layering for High Speed Communication：the MPI over Fast Message (FM) Experience. Cluster Computing，1999，2(2)：107～116

5 Jacunski M，et al. Low Latency Message-Passing for Reflective Memory Networks. In：CANPC'99，LNCS 1602，1999. 211～224

6 von Eicken T，et al. Active Messages：a mechanism for integrated communication and computation. In：Proc. of the Intl. Symposium on Computer Architecture，1992. 256～266

7 Rodrigues S，Anderson T，Culler D. High-performance local-area communication using Fast Socket. In：Proc. of the USENIX 1997 Technical Conf. San Diego，California，USENIX Association，1997

8 Yocum K G，et al. Cut-through delivery in Trapeze：an exercise in low-latency messaging. In HPDC-6，Portland，Oregon，August 1997

9 von Eicken T，et al. U-Net：A user-level network interface for parallel and distributed computing. In：Proc. of the 15th ACM Symposium on Operating Systems Principles. Dec. 1995. 40～53

10 Dubnicki C，et al. VMMC-2：efficient support for reliable，connection-oriented communication. In：Proc. of Hot Interconnects V. IEEE，Aug. 1997

11 Tezuka H，Hori，A，Ishikawa Y. PM：A high-performance communication library for multi-user parallel environments：[Technical Report TR-96-015]. Tsukuba Research Center，Real World Computing Partnership，Nov. 1996

12 Prylli L，Tourancheau B. Protocol design for high performance networking：a Myrinet experience：[Technical Report N. 97-22]. LIP，Ecole Normale Superieure de Lyon，July 1997

13 Leffler S J，McKusick M K，Karels M J，Quarterman J S. The Design and Implementation of the 4.3 BSD Unix Operating System. Addison-Wesley Publishing Company，Inc.，Redding，Massachusetts，1989

14 董迎飞. 分布式存储多机系统中通信技术的研究：[清华大学博士学位论文]. 1995

15 陈国良. 并行计算-结构. 算法. 编程. 高等教育出版社，1999

16 见网址：http://www.myri.com/myrinet/overview/index.html. Myrinet Overview

17 见网址：http://www.myri.;com/scs/index.html. The GM API

18 申俊，郑纬民，王鼎兴，沈美明. 提高工作站机群系统通信性能方法的研究. 小型微型计算机系统，1997，18(6)：8～13

（上接第 4 页）

结论和展望 ACI 试验床通过高速网络将分布在各地的超级计算机连接起来，形成整体高性能计算能力，为各个地方、各个学科专业的用户提供高性能计算服务，促进高性能计算在国内教育与研究领域的应用。

由于 ACI 的根本目的是推动高性能计算在我国的普及，为此我们在系统与用户的接口方面做了大量的工作。一方面把整个系统的使用环境建立在 Web 之上，使得用户可以通过 Internet 提交、观察和控制任务并得到运行的最终结果；另一方面，我们大量使用了可视化技术，以直观简洁的方式反映运行的状态和结果，消除非计算机专业人员使用高性能计算机的障碍。不同学科的科技人员，已经利用 ACI 完成了一些实际课题的研究工作。它充分体现出了这个平台的优越性。

我们今后的工作是将 ACI 系统建设成为一个全国范围的虚拟机房，建立共同的安全与授权标准、分布式调度、计算模块库和计费管理系统，提供海量存储能力和更加方便的用户接口等。随着网络由窄带向宽带、有线向无线的扩展，以及内嵌处理器、传感器和摄像机等各种先进设备的不断加入，ACI 系统平台将进一步成长，必将推动高性能计算技术在宇宙学、天文物理学、地球科学、流体力学、工程学、生物学和化学等更多学科中的应用。

参 考 文 献

1 ASCI Project. http://www.lanl.gov/projects/asci/asci.html，http://www.llnl.gov/asci/，http://www.sandia.gov/ASCI/，http://www.sandia.gov/ASCI/，http://www.lanl.gov/asci/，http://www.lanl.gov/projects/asci/asci.html

2 Grand Challenges：High-Performance Computing and Communications. Committee on Physical，Mathematical，and Engineering Sciences，National Science Foundation，Washington，D. C.，1991

3 Bell G. Ultracomputers：A Teraflop before Its Time. Communications of the ACM，Aug. 1992. 27～47

4 Foster I. Internet Computing and the Emerging Grid. Nature，7 December 2000. http://www.nature.com/nature/webmatters/grid/grid.html

5 Foster I，Kesselman C. Computational Grids：The Future of High-Performance Distributed Computing. Morgan Kaufmann Publishers，1998

6 Gutmann M J. Cluster PCs for Power. BYTE，July 1993. 57～64

7 Li Sanli，Li Yamin，et al. Q-Net：A Fast Interconnection Network Switch for NPCS-1 Parallel Systems. In：The 10th Intl. Conf. of Parallel and Distributed Computing Systems，Las Vegas，Nevada，USA，2000. 28～31

8 The Java 3D API Specification. Sun Microsystems，Inc.，2000