文章编号:1671 - 6833(2006)01 - 0067 - 05

广域网协议在 PC 集群系统 HPC 中的应用分析

王文义[1],赵少林[2],王若雨[3]

(1.中原工学院计算机系,河南 郑州 450007 2.郑州大学信息工程学院,河南 郑州 450052 3.河南电力职工大学网络中心,河南 郑州 450051)

摘 要:鉴于绝大多数 PC 集群系统都使用了 TCP/IP 协议,着重分析了作为分布式进程间通信手段的 socket 通信机制在 Linux 中的实现以及传统通信开销中影响性能的主要因素,并针对通信瓶颈,对传统的通信协议栈进行了改进,对广域网协议在 PC 集群系统中应用的不足之处,提出了改进集群网络性能的方法.

关键词:集群系统;高性能计算;TCP/IP

中图分类号:U 448.225 文献标识码:A

0 引言

一般集群系统是指用高速网络技术将一组高性能工作站或微型计算机以某种结构连接起来,在并行程序开发环境支持下统一调度的分布式存储处理系统.它主要通过消息传递方式实现各结点机之间的通信,并由建立在操作系统之上的并行编程环境来完成对系统的资源管理及协同工作,同时也屏蔽工作站及网络的异构性.PC 集群系统以其低成本、高性能的特性,提供了强大的批处理和并行处理能力,越来越受到广大普通用户的欢迎.

1 Linux 操作系统的 TCP/IP 结构

linux 对 IP 协议族的实现机制如图 1 所示.像网络协议自身一样,linux 也是通过把它看作为一组相连的软件层来实现的[1].其中 BSD 套接字(BSD Socket)由通用的套接字管理软件所支持,它是一个通用的系统接口,不仅支持各种形式的网络连接,同时也是一种进程间的通信机制.套接字可以描述通信连接一端的运行状态,对于参与通信的两个不同进程有不同的套接字与之对应.在 Linux 中,BSD 套接字是作为属于特殊文件系统的文件来实现的.BSD 套接字的数据结构和一些重要的域:

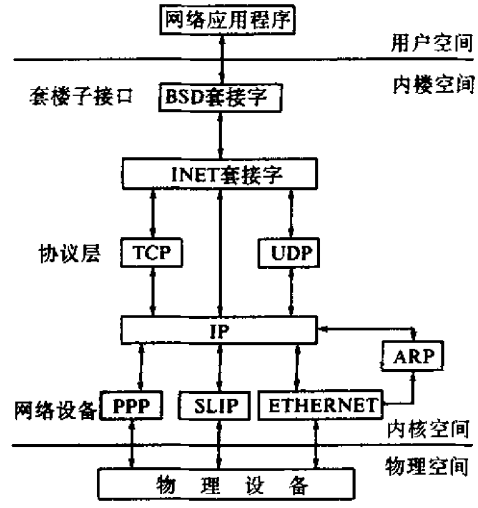

图 1 Linux 系统中 TCP/IP 的层次结构

Fig.1 The TCP/IP hiberarchy of Linux system

```
struct socket {
    socket_state state;
    unsigned long flags;
    struct proto_ops * ops;
    struct inode * inode;
    struct fasync_struct * fasync_list;
    struct file * file;
    struct sock * sk;
    wait_queue_head_t wait;
    short type;
    unsigned char passcred;
};
```

收稿日期:2005 - 08 - 14 修订日期:2005 - 10 - 08

基金项目:国家重点新产品计划项目(2002ED782017).

作者简介:王文义(1947 -),男,河南省洛阳市人,中原工学院教授,主要研究方向为高性能计算机集群系统及其并行处理技术.

inode:指向 sockfs 的 inode 的指针.

file:指向 sockfs 文件的文件对象.

state:存储 socket 的连接状态:SS_FREE(未分配),SS_UNCNNECTED(未连接),SS_CONNECTING(正在连接),SS_CONNECTED(已连接),SS_DISCONNECTING(正在断开连接).

ops:指向一个 proto_ops 数据结构,存储的是 socket 的方法,大多数方法引用系统调用对 socket 进行操作.每一个网络架构都有自己的函数用来实现自己的方法;因此,同一个系统调用针对不同的网络架构实现的功能不尽相同.

sk:指向数据结构 sock,用来描述底层的套接字.

描述套接口通用地址的数据结构 struct sockaddr:

```
struct sockaddr {
    sa_family_t sa_family; /* address family, AF_xxx */
    char sa_data[14]; /* 14 bytes of protocol address */
};
```

sock:

sock 数据结构有将近 80 个域,其中大部分是指向其它对象、方法列表或数据结构的指针.每个 sock 数据结构维护一个关于 BSD socket 的协议信息.例如,对一个 INET 类型的 socket,其数据结构将含有所有 TCP/IP 和 UDP/IP 的相关信息.

Socket 类型决定了网络通信的风格.TCP/IP 中的 Socket 有 3 种:

(1) 流式 Socket (sock_stream).流式套接字提供可靠的、面向连接的通信流;它使用 TCP 协议,保证了数据传输的正确性和顺序性.

(2) 数据报文 Socket (sock_dgram).数据报文套接字定义一种无连接的服务,数据通过彼此独立的报文进行传输,它是无序的且不保证可靠、无差错.它使用数据报协议 UDP.

(3) 原始 Socket.原始套接字允许对底层协议如 IP 或 ICMP 直接访问,功能强大但使用较多限制,主要用于一些协议的开发并需要有超级用户权限.

2 影响通信性能的因素

衡量集群网络性能的重要因素是通信延迟时间,它包括协议软件处理开销和网络硬件处理时间.集群系统的通信性能是整个系统的瓶颈,其主要涉及四个因素:网络带宽、操作系统的额外开销、TCP/IP 协议对网络性能的影响以及协议中复杂的缓冲管理[2].笔者着重讨论 TCP/IP 协议对网络性能的影响.

2.1 TCP/IP 协议对集群系统网络性能的影响

2.1.1 TCP/IP 协议多层次结构引发的处理开销

目前的集群系统大多采用 Linux 操作系统.TCP/IP 协议是面向低速率、高差错和大数据包传输而设计的,是一个多层次的软件结构.该结构按自底向上的顺序可分为四层:网络接口层、网间网层(IP)、传输层(TCP)和应用层[3].在进行数据传输时,由于数据需要经过多次拷贝才能完成从应用层到网络接口层的传递(反之也一样),从而带来了较大的网络延迟开销.统计表明,每增加一次拷贝,通信的效率就会降低至少 20%[1].另外,在多层协议的实现中,有很多相同的功能需要在各层重复实现,比如:①从 IP 层到传输层都要进行差错控制;②从网络接口层到应用层都要进行协议的处理机调度;③从 IP 层到应用层都要进行流量控制;④从 IP 层到应用层都要进行数据包组装和定序的缓冲;

2.1.2 TCP/IP 协议在集群系统中应用的不足

众所周知,TCP 只支持一对一的对等连接(一个发送者对一个接收者)而不具备组播的功能.因此,对于一个发送者需要有 10 个接收者的通信就需要有 10 次连接和 10 次独立的传输.在集群计算中,由不同结点机的多个应用程序共享同一数据流的情况是较为普遍的.虽然 UDP(用户数据报协议)可以利用组播,但它是不可靠的服务.另,对于发送和接收实时数据的集群应用程序至关重要的优先级和延迟控制,TCP 提供的支持也是不够的.

TCP 只支持完全可靠的传输,这对于某些应用程序来说显得过于强壮,而只支持"最大努力"服务的 UDP 却又显得可靠性不够.数据可靠性是由协议而不是应用程序定义的,因此对集群应用程序应该具有可以在连续连接的基础上定义自己可靠性的范例.为了可靠地发送单条消息,TCP 要求交换六个分组(两个用于建立和确认连接,两个用于发送和确认数据,两个用于关闭连接)并在实际上是分别完成的,这对于生命周期较短的连接来说减慢了数据的交换,而集群用户的应用程序却往往对处理事务的速度有很高的要求.

TCP、UDP 都实现了协议定义而非应用程序所定义的固定策略.尽管这是正确的,但由于所有

的连接都由流控制所支配,因此集群系统中的一些不受限制的“流”应用程序就不能获益.其关键是 TCP 和 UDP 将机制嵌入了策略,两者无法分开.对于高性能集群计算来说,应该给协议提供机制,为分布式应用程序建立相应的策略.

3　提高和改进集群通信性能的途径与方法

3.1　提高通信性能的主要途径

通信开销在很大程度上是由于协议层次多、数据拷贝频繁引起的.由于通用的网络通信协议为了满足各种不同用户的需求,增加了诸多与数据传输无关的服务性开销但却以降低通信性能为代价.

使用更高性能的网络结构可以直接提高集群的通信性能,但这时设备的价格也往往较高,而且由于目前高速网络的运用,使得影响通信性能的瓶颈已从过去的网络硬件转移到了网络通信软件上.另,减小高速网络小数据包传输的延迟时间也是提高集群通信性能的途径之一.

3.2　改进集群通信性能的方法

集群系统通过 TCP/IP 协议层进行访问,虽然方便但是性能较低;直接访问硬件又会增加编程难度,并且程序的通用性也差[4].因此考虑既可以不使用 TCP/IP 协议,通过直接与以太网适配器进行通信,又能屏蔽许多低层的操作和控制的方法,这样就可以为程序员提供一个比较通用性和透明性的保证.由于 HPC 系统通常运行在局域网下,物理层的可靠性非常高[5],信道出错概率达到 10^{-15},因此在新协议中不考虑查错及纠错等功能,只进行必要的错误处理即可,这样,就可以大大提高集群系统的通信性能.

在 Linux 内核中,SOCKET 套接字提供了一个特定类型关键字 SOCK _ PACKET,只有超级用户(用户注册号为 0)的进程才有权使用它.可以通过创建 PACKET 类型的套接字,让应用程序直接在数据链路层接收或发送原始数据报文以满足要求.

Socket( )的功能是通过内核函数 sys _ socket( )具体实现的,其步骤如下:

(1) 为新的套接字分配一个描述符.内核在特殊文件系统 sockfs 中创建一个 inode.BSD 套接字的属性存储在 socket 类型的结构中,它是 u.socket _ i 中的一个对象,u.socket _ i 是 sockfs 文件系统的 inode 中的文件系统说明域.

(2) 根据指定的协议族、套接字类型和通信协议初始化 D 套接字.相对于 TCP/IP 协议族的初始化函数是 inet _ create( ),它首先检查由 socket( )函数的参数所指定的通信模型和协议是否与 TCP/IP 协议族兼容,然后再分配和初始化一个新的 INET 套接字,并把它和套接字相连接.

(3) 分配一个新的文件描述符和文件对象,并把文件描述符和套接字中相应的指针域赋值为文件对象的地址,实现三者的连接.

socket( )系统调用在两个进行通信的进程之间创建了一个通信端口,调用成功后返回一个文件描述符.其实,一个套接字跟一个打开的文件管道非常相似,它也允许通过系统调用 read( )和 write( )来读取和写入数据,只不过数据的来源和目的地都出自于另外一个进程.

利用 int socket(int domain,int type,int protocol)函数调用创建套接字.其中,参数 domain 指定要创建的套接字的协议簇;参数 type 指定套接字类型;参数 protocol 用来指定 socket 使用的传输协议编号.其中指定 type 为 sock _ packet 类型,具体传输协议编号由调用时给出.若 socket 调用成功则返回 socket 处理代码,失败则返回 - 1.

sockfd = socket(PF _ INET, SOCK _ DGRAM, 0),表示使用的是 TCP/IP 协议族和非面向连接的数据包套接字,通过前两个参数可定位为 UDP 协议,第三个参数 0 表示由前两个参数自动决定使用的协议[6].

数据包采用 IEEE802.3 标准帧结构,其基本格式为:

帧头	起始定界标志	目的地址	源地址	数据长度	数据	帧校验序列

在帧结构中,数据域的长度是可变的,而其它域的长度则是固定的.在 SOCK - PACKET 类型的套接字调用时,帧头、帧的起始定界标志以及帧校验序列都是由 SOCKET 调用自动添加的.因此,在应用时,只需要填写目的地址、源地址、数据长度和数据即可.应该注意的是,在以太网中最小的数据长度为 46 字节,最大不能超过 1 500 字节.如果待发送数据小于 46 字节,应填充一些无用数据,以达到要求,同样,如果数据超过 1 500 字节,要将其分成若干个帧发送.

当向网络发送数据包时,可以调用如下函数:

```
Int sendtonet(int sock,const char * device,
const char * data,int len)
{stuct sockaddr sendandrec;
Sendandrec.sendandrec _ family = AF _ IN-
ET;
```

```
Strcpy(sendandrec.sendandrec_data,device);
Return (sendto (sock, data, len, 0,
&sendandrec,sizeof(sendandrec)));
}
```

其中,data是一个指向待发送数据的指针.应注意的是,data指向的数据首先是目标地址,然后才是地址和数据长度,最后是数据通信.由于是直接与网络适配器进行通信,这些信息必须自己填写.

当从网络接收相应的数据包时,可以调用如下的函数recfromnet:

```
Int Recfromnet(int sock,char * device,char
* data,int len)
{struct sockaddr sendandrec;
Int reclenth = sizeof(sendandrec);
Int error;
Error = recvfrom (sock, data, len, 0,
&sendandrec,&reclenth);
If(error = = -1)
Return -1;
Strcpy(device,sendandrec,sendandrec_da-
ta);
Return error;//Actually size of received
packet.
}
```

接收无误后,返回的是实际接收到的数据字节数,否则,返回错误信息-1.在发送和接收数据操作完成后,应该调用函数关闭SOCKET套接字.通过上述函数调用,可以避免TCP/IP的巨大开销而初步实现低层通信的基本功能.

发送进程:

(1) 数据准备,发送一组报文(报文总数确定),依次发送1 500字节的报文,直到该报文发送完毕.设定重试次数并启动超时定时器;

(2) 发送进程等待接收进程发来的确认信息或者超时;

(3) 若收到确认信息且未超时,那么检查是否信息丢失,若没有,则发送下一分组,转(1);否则重新发送相应的信息,转(2);

(4) 若发送进程超时,则重新发送最后一个报文,重试次数为 $N = N - 1$,若 $N = 0$,转(5),否则转(2);

(5) 重试 N 次失败,向上一层报告错误信息.

接收进程:

(1) 当收到一组最后一个报文(报文总数确定)或者到达用户数据尾端时,检验是否有数据包丢失,若没有则将相应的屏蔽字置1,否则置0.如果有丢失,则向发送进程请求重发该包.启动超时定时器,并置重试次数为N,等待接收对应的数据包.如果没有丢失,则等待接收下一分组.

(2) 若再次收到一组最后一个报文或用户数据尾端,需再次向发送进程请求重发数据包,并置重试次数 $N = N - 1$,若 $N = 0$,转(4),否则启动定时器,转(1).

(3) 若超时,置重试次数 $N = N - 1$,若 $N = 0$,则转(4),否则,发送一个重传请求,启动定时器;

(4) 重试N次失败,向上一层报告错误信息.

Socket接口技术已在许多分布式系统应用中使用.通信库如MPI和PVM可在socket之上实现.公共协议结构如图2所示.

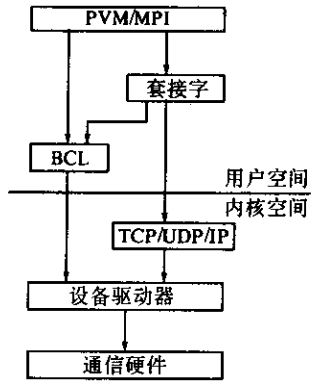

图2 公共协议结构

Fig.2 structure of common protocol

在通信过程中,对于发送方,消息向下传输到socket层、TCP/IP(或UDP/IP)层及驱动器和网络硬件层;对于接收方,则是以逆序重复同样的传输过程.Socket技术可以直接在底层的基础通信层(BCL)上实现,绕开TCP/IP协议族而直接与设备驱动程序通信,以达到节省开销、提高集群通信效率的目的;另外,也可以通过避开Socket/TCP/IP层,在BCL之上实现PVM/MPI.BCL的主要目的是让用户尽可能多的开发原始硬件的性能,这也是当前研究的热点和未来集群通信技术发展的一个趋势.

4 结束语

事实证明,专为广域网设计的TCP/IP协议,在局域网集群系统中的应用却未必是高效的.因此,高性能集群系统也需要有自己专用而瘦身的通信协议.Linux系统中预留的SOCK_PACKET套接字为此提供了实现的途径,可以充分利用它来

设计开发专用的通信协议,以提高集群系统的通信效率.通过精简网络协议层次,在 Linux 集群系统中旁路 TCP/IP,减少了套接字的多次调用和重复拷贝.并针对局域网的特点,只进行必要的差错控制.使用选择重传算法保证传输可靠性,缩短通信延迟,降低了通信软件的开销,以此换取了更好的性能.

参考文献:

[1] 黄晨春,田艾平. Linux 的 TCP/IP 层结构[J].计算机应用研究.2002,5:127 ~ 129.

[2] 申　俊,郑纬民,王鼎兴,等.提高工作站机群系统通信性能方法的研究[J].小型微型计算机系统.1997,18(6):8 ~ 13.

[3] [美]RICHARDSTEVENS W.TCP/IP 详解(卷 1):协议[M].范建华,胥光辉,张　涛,等译.北京:机械工业出版社,2000.

[4] 罗四维,王　祯. 机群系统中的简单可靠协议通信技术研究[J].北方交通大学学报.2003,27(5):1 ~ 6.

[5] 吴文峻,向晓华,龙　翔.高速机群互连网络链路层协议设计[J].北京航空航天大学学报.1998,24(4):458 ~ 461.

[6] 徐千洋. Linux C 函数库参考手册[M].北京:中国青年出版社 2002:342 ~ 343.

Analysis of the Application of The Protocol of WAN in the PC Cluster Computer System HPC

WANG Wen – yi[1], ZHAO Shao – lin[2], WANG Ruo – yu[3]

(1. Department. of Computer Sience, Zhongyuan Institute of Technology, Zhengzhou 450007, China; 2. School of Information Engineering, Zhengzhou University, Zhengzhou 450052, China; 3. Network Center, Henan University of Electric Power and Workers, Zhengzhou 450051, China)

Abstract: Currently, most PC cluster systems use TCP/IP as the communication protocol. So this paper analyses the mechanism of socket communication between distributed machines in the Linux kernel. The main factors affecting the communication performance in PC cluster system are analyzed in detail. The structure of communicatin protocol stack is ameliorated aiming at the bottleneck. The paper discussed the use of TCP/IP in PC cluster systems, finds out the inapplicability of the TCP/IP in systems and proposes a new way to improve the communication performance.

Key words: cluster computing system; HPC; TCP/IP

郑州大学第十批博士点增幅大

刚刚从国家教育部网站上获悉,1 月 25 日教育部下发学位[2006]3 号文件《关于下达第十批学位授权学科专业名单的通知》指出,第十批博士和硕士学位授权学科、专业名单,已经国务院学位委员会第 22 次会议批准,在文件公布的第十批博士和硕士学位授权学科、专业名单中,郑州大学历史学、物理学、化学、材料科学与工程、水利工程、化学工程与技术、基础医学、临床医学 8 个学科成为一级学科博士学位授权点,宪法学与行政法学、思想政治教育、中国古典文献学、化工过程机械、通信与信息系统、控制理论与控制工程、计算机软件与理论、防灾减灾工程及防护工程、劳动卫生与环境卫生学、药物化学 10 个学科获得二级博士学位授权点,获得硕士学位一级学科 34 个,获得硕士学位授权学科专业 19 个。