基于总线桥协议的异构机群并行虚拟机的构造*

金利杰 张建军 李未

(北京航空航天大学计算机系 北京 100083)

摘要 BBP_PVM是为北京航空航天大学计算机系基于总线桥协议的异构可扩展并行计算机群系统BBP_SPC(bus bridge protocol-scalable parallel computer)研制的PVM版本. BBP_PVM以总线桥多机互联协议的消息传递层子协议(BBP_MPL)为虚拟机内各处理机间的通讯协议. BBP_MPL是在BBP可靠链路的基础上实现的精简和可靠的机间通讯协议,BBP_MPL的采用有效地降低了通讯过程中报文应答、重发和动态缓冲区管理的开销. BBP_PVM与PVM3.3.4及其以上版本兼容.

关键词 总线桥多机互联协议,精简通讯协议,可扩展并行计算机群,并行虚拟机.

中图分类号 TP302

1 BBP_PVM的研制目标

可扩展并行计算机群构造技术研究和样机系统研制的目的是寻找有效的技术手段,通过灵活地聚合高性能单处理机的能力,实现并行计算机规模和能力与应用算法间的协调. 当系统规模扩展时,系统的处理能力应与处理机台数的增加而线性提高. 当问题规模扩大时,并行计算系统的规模和性能均应能够以较小的成本,以较为便利的方式扩展.[1]

由于我国当前VLSI技术的工业基础薄弱,工艺水平较世界先进水平有差距,设计及验证技术、装备和手段也较贫乏,自行设计先进处理机芯片及其周边系统十分困难. 采用总线桥多机互联协议连接商品化的高性能PC机和工作站,构成规模可扩展的并行计算系统[2],是符合我国当前技术水平的发展道路之一.

BBP_SPC(bus bridge protocol-scalable parallel computer)是北京航空航天大学研制的基于总线桥多机互联协议(Bus Bridge Protocol)[3,4]的异构可扩展并行计算机群系统. BBP_SPC的设计思想是:从高档微机和工作站的系统总线入手,采用总线桥多机互联协议和适当的实现技术,以较低的成本,获得较高的数据传输速率链路速率,在此基础上构造可

* 本文研究得到国家863高科技智能计算机主题、航空科学基金和国家博士后科学基金的资助. 作者金利杰,1965年生,副教授,主要研究领域为并行处理,体系结构,分布式共享存储. 张建军,1972年生,硕士生,主要研究领域为并行软件技术. 李未,1943年生,教授,博士导师,主要研究领域为计算机理论,软件开发环境,计算机应用.

本文通讯联系人:金利杰,北京100083,北京航空航天大学计算机系

本文1996-06-11收到修改稿

扩展并行机群系统.[5]

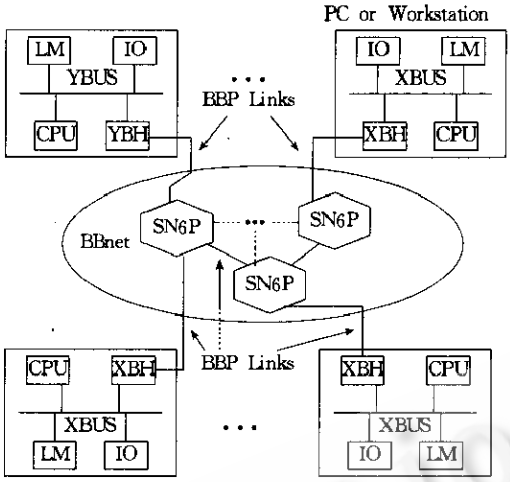

图1 BBP_SPC结构示意图

BBP_SPC 样机系统采用 PentiumPC 和 SunSPARC 工作站作为结点处理机. 系统包含的处理机数目为 8 台,其中 Pentium-133Mhz PC 机 6 台,Pentium PC(90MHz)1 台,SunSPARC station-II 工作站 1 台. 它们通过总线桥多机互联网(BBnet)连成计算机群. BBP_SPC 系统的结构如图 1.

总线桥多计算机互联协议 BBP 是设计和实现可扩展机群互联系统 BBnet 的依据. BBP 作为一个中间协议,只与所选计算机型的内部总线标准有关,而与其处理器的类型无关,从而增强了系统体系结构及互联系统的生命周期和适应能力.

BBP_SPC 系统连接技术的核心是多端口互联器(SN6P)构成的总线桥互联网—BBnet.[2,3,6,7]BBnet 支持处理机间的高效信息通信,它的单链路数据通过能力峰值为 60MBytes/s. BBnet 通过将多个 SN6P 直接连接实现规模的扩展. 每一 SN6P 可同时支持 3 条 BBP 链路的并发操作,聚合数据通过能力为 180 MBytes/s.

标准总线协议 XBus 至 BBP 协议的转换装置称为 XBH. 如 SBus 总线至 BBP 协议的适配器称 SBH,PCI 总线至 BBP 协议的适配器称 PBH. 由 PBH 组成的 BBP 链路峰值传输速率为 18MBytes/s,典型传输速率为 6.98MBytes/s.

PVM 是美国橡树林国家重点实验室、田纳西大学等单位开发的以计算机网络技术为主要基础的并行应用软件开发环境. BBP_PVM 以 PVM (3.3.4 版)为基础实现,其目标是成为 BBP_SPC 上开发并行计算软件的支撑平台. BBP_PVM 以精简可靠的 BBP 多机互联协议为消息传递的基础,能够降低通讯过程中消息应答、重发和缓冲区管理的开销. BBP_PVM包含的 Flexible Message(能动消息)机制,能够充分重叠处理机的计算过程和消息传递过程,有效地降低通讯延迟. BBP_PVM 的共享设备机制使原来 pvmd 承担的部分通讯控制工作交由 BBP_PVM 的启动和运行控制进程 bbp_demo 完成,pvmd 可以更有效地管理处理机间的消息传递. BBP_PVM 依托 BBP 的并发链路机制,能够实现有效的并发消息队列管理,降低由软件引起的通讯延迟.

2 BBP_SPC 的软件体系

BBP_SPC 的并行程序开发软件环境建筑在 BBP_MPL 之上,它包括并行环境的支撑平台 BBP_PVM、可视化工具 BBP_ParaTools、并行程序库 BBP_ParaLib 和一些典型的应用程序.

BBP_PVM 的消息收发命令由每台处理机控制其 XBH 完成. BBP_MPL 消息传递层子协议以一组函数的形式向 BBP_SPC 的并行系统软件和应用软件提供处理机间的连接和

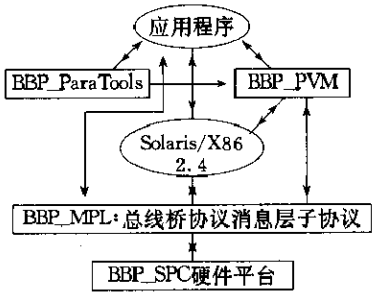

图2 BBP_SPC的软件体系

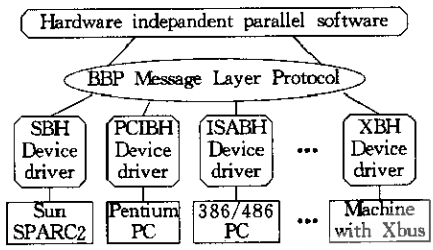

图3 BBP_MPL的功能示意图

传输服务,这些服务得到基于总线桥多机互联协议的 BBnet 有效支持. BBP_MPL 通讯服务函数由 BBP_MPL 数据结构、BBP_MPL 底层服务函数和 BBP_MPL 高层服务函数组成.

2.1 BBP_MPL 数据结构

BBP_MPL 的主要数据结构为:

```
struct xbh_unit {
        u_char unit_open;
        u_char xbh_busy;
        int xbh_state;
        int p_pgrp;
        ......
}
u_char linked;
```

其中(1)*xbh→unit_open*:当XBH 板被打开时,*xbh→unit_open*=1;当XBH 板关闭时,*xbh→unit_open*=0;

(2) *xbh→xbh_busy*:当前设备处于工作状态时,*xbh→xbh_busy*=1;否则 *xbh→xbh_busy*=0;*xbh→xbh_busy* 用于限制了多个进程同时读写同一端口时,消息间的互相屏蔽.

2.2 BBP_MPL 的底层函数

BBP_MPL 的底层函数实际上是 XBH 的 UNIX 设备驱动程序的接口函数,它包括 *xbh_open*(),*xbh_close*(),*xbh_read*(),*xbh_write*(),*xbh_ioctl*()等系统调用入口. 结点处理机通过调用这些函数可获得通讯许可和对通讯系统状态的判断,完成机间消息传递.

函数 *xbh_open*()的主要任务是负责给 XBH 分配设备标识句柄;设置资源及状态互锁;设定 XBH 设备的状态为设备打开且空闲;设定 XBH 的 BBP 链路端口空闲;根据调用参数确定通讯模式;标识当前操作 XBH 的进程号;复位接口设备状态等.

函数 *xbh_close*()的主要任务是在最近一次通讯过程结束后,关闭设备;清除 XBH 操作进程号;撤销 XBH 的 BBP 链路连接;关闭资源及状态互锁.

函数 *xbh_read*()在设备状态和链路状态满足设定的条件时,从 XBH 的 BBP 端口读取消息,然后交付于等待该消息的进程.

函数 *xbh_write*()在设备状态和链路状态满足设定的条件时,向 XBH 的 BBP 端口输出消息.

函数 *xbh_ioctl*()完成一系列由调用参数指定的功能,包括设备状态查询、BBP 链路建

立和撤销、BBP 链路测试、XBH 寄存器操作、设备复位、单步通讯等等.

2.3 BBP_MPL 高层函数

BBP_MPL 高层函数包括 *mesg_dev_open*(*style*, *mesg_handler*), *mesg_dev_close*(), *mesg_send*(*sit_no*, *message*, *size*, *flag*), *mesg_handler*(), *mesg_block* 和 *mesg_unblock*().

mesg_dev_open(*style*, *mesg_handler*)和 *mesg dev close*()是提供给 BBP_PVM 通讯管理软件的消息设备开关函数. 对这2个函数的调用通过操作系统变换为对 *xbh_open*()和 *xbh_close*()的调用.

mesg_send(*sit_no*, *message*, *size*, *flag*)向 *sit-no* 指明的目标地址发送保存在缓冲区 *message* 中的长度为 *size* 的消息包. 调用成功时返回实际发送的消息包长度.

mesg_handler()响应来自 XBH 的消息报文到达中断,从读取消息报头中读取目的地址、正文长度、消息报文队尾标志,放于消息报接收队列 SPQ;

mesg_block()和 *mesg_unblock*()负责关键区的管理. 在 BBP_PVM 中,消息报文输出和输入队列均由双向链表组成,对报文的操作通过对指针的操作实现. 例如,当 XBH 的报文到达中断时,BBP_PVM 触发一个新的 *mesg_handler* 负责接收消息报文,并将其放于报文接收队列 SPQ 尾部,这一操作涉及到双向链表指针操作. 如果复杂指针操作被中断,新接收到的报文将放于未操作完成的链表中,会导致指针指向非法访问区. 因此,BBP_PVM 通过关键区的设置和管理来实现消息报文队列指针等关键资源操作的互斥.

3 BBP_PVM 的结构

BBP_PVM 由4部分组成:启动控制进程 bbp_demo;PVM 守护进程 pvmd;用户库;BBP_MPL 通讯原语.

3.1 bbp_demo

运行于每台处理机之上的 bbp_demo 负责启动远程 pvmd 和维护 BBP_PVM 系统配置. 每台处理机开机时都运行一个 bbp_demo,它们的地位平等. BBP_PVM 修改配置时,主 pvmd 向从方处理机发出修改配置的控制命令;由从方 bbp_demo 根据消息参数启动一个 slave pvmd,并提供一个与 slave pvmd 的连接,然后从方 bbp_demo 转入循环响应远程或本地的控制命令.

3.2 BBP_PVM 的守护进程 pvmd

pvmd 运行在虚拟机的每个结点处理机上,作为消息路由器和控制器,提供任务分配和进程控制. 当应用程序结束,pvmd 仍运行在机群的每台处理机上,维持 BBP_PVM 系统.

首先启动的 pvmd 可以被看作是主 pvmd,其后启动的 pvmd 作为从 pvmd,多数情况下,它们是平等的. 但主 pvmd 可以启动新的 BBP_PVM 结点机,修改系统配置. pvmd 中最重要的数据结构是 host table 和 task table,它们分别用来描述当前虚拟机的系统配置和任务运行状态,并维护多任务通讯队列. 表中每一个 host 和 task 都附有 pkt 和 mesg 队列,分别存放着进出不同目的机和不同任务的包和消息.

pvmd 间的通讯协议以总线桥多机互联协议 BBP 为基础. BBP 在 pvmd 间提供可靠的通讯链路,能够保证消息报文不丢失,并能自动维持报文序列,在 pvmd 间通讯的软件管理

层次上无需重发控制机制和报文重组机制,大大简化了原版 PVM 中的通讯管理,降低了由通讯软件引起的开销.

pvmd 中维持一个消息接收队列 spq,同时为发往不同机器的消息分别建立一个发送队列 hd_txq.

在发送时,pvmd 轮流判断每个发送队列是否为空. 如果存在不空队列,则申请与队列对应的处理机建立 BBP 链路. 申请成功,则将这个队列上的消息全部送走,然后撤消该链路. 若在发送前链路申请失败,则等待下一次链路申请.

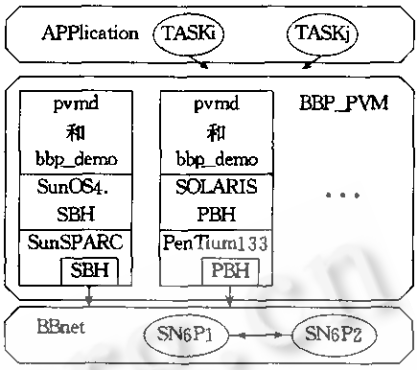

图4 BBP_PVM结构图

发送和接收队列的消息缓冲区均动态分配和释放. 当 pvmd 收到其他的 pvmd 的链路连接请求时,它激活一个 handler 接收这条链路上发来的消息,并存入接收队列 spq.

pvmd 根据启动参数初始化为主 pvmd 或从 pvmd 后,进入如下过程:

```
打开 XBH 设备;
进入主循环 work( ) {
    if (子任务死亡)
        将分配给子任务的资源收回,并广播通知所有试图与该任务通讯的进程;
    netoutput ( )
        为发向不同 pvmd 的 pkt 依次申请链路;在一次链路中,发送完所有到该 pvmd 的 pkt;
    netinput ( )
        处理从互联开关上接收来的 pkt;
    select( ) 查询本机内各 socket( )端口;
    if (有socket 端口申请链接)
        loclconn( );
    for (locltasks) {
        if (socket 端口读就绪)
            loclinput();
        if (socket 端口写就绪)
            locloutput();
        if (socket 端口输出就绪)
            loclstout();
        } \end of for\
} \end of work( )\
```

在 pvmd 中每个消息报文包有16字节的报头,报头格式为

Dest TID						
Source TID						
Seq Number						Ack Number
	A	F	D	E	S	
...	C	I	A	O	O	UNUSED
	K	N	T	M	M	

图5 pvmd 报头格式

其中 Dest TID 和 Source TID 分别指明接收和发送任务的 TID 号.

在 pvmd 中,每个消息都由一个以上的报文组成,当 PVM 接收到新的报文,根据报头信息处理每个报文. SOM 指明本报文为一个消息的第1个报文;EOM 指明本报文为一个消息的最后1个报文;DAT 指明该报文为数据报文,而非控制报文;~DAT 指明该报文为非数

据报文.

图6描述了整个BBP_PVM系统的当前机器配置,包括每个节点机的体系结构、运算速度、通讯带宽、任务号、发送和接受消息队列以及每个host的当前消息进出状态.当pvmd派生一个子任务时,根据这张表选择一个负载较小的host启动新的任务,可以保持系统负载平衡.

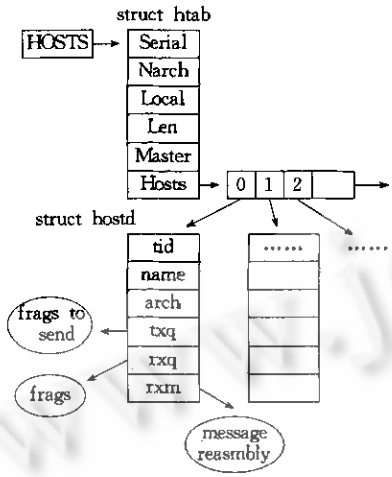

图6 BBP_PVM的内部配置

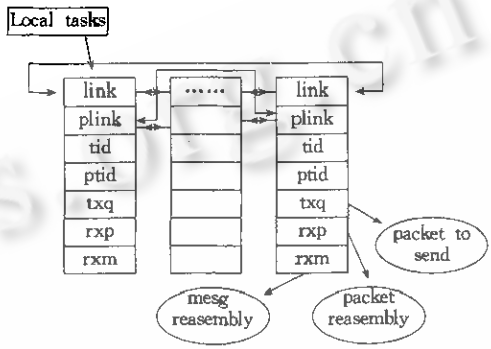

图7 BBP_PVM的任务表

pvmd为发向不同结点机的每个报文进出队列动态申请链路,动态维护和缓冲消息.同时,每个pvmd都维持一张当前任务表,每个任务分配一个任务指针,且具有唯一的tid号;pvmd为每个任务维护一个报文进出队列.

BBP_PVM的任务管理基于UNIX的进程管理.每个pvmd管理所在处理机的任务,用户的并行任务分布在各个结点机上,由BBP_PVM统一管理.

图7为pvmd的任务管理表,每个pvmd的任务队列由一个双向链表构成.其中txq为报文发送队列,pvmd通过TCP的socket()机制与本地的任务交互;rxp为报文接收队列;rxm为由rxp队列中的报文组成的消息队列,并根据消息类型,发出相应操作.

4 BBP_PVM的关键技术

基于总线桥多机互联协议的BBP_PVM的实现方案中,采用了多种对系统性能有明显改善作用的关键技术.

4.1 能动消息通讯模式

采用传统的阻塞通讯模式,BBP通讯链路将成为瓶颈.在一次信息传输前,通讯双方将占用一条BBP链路和连接这条BBP链路上的所有端口.如果发送方已经发送数据,而接收方没有立刻读走数据,整个BBP链路将阻塞,并且发送方不能返回,直到接收方读走数据,BBP链路才会释放,相应的BBP端口也才可重新被申请使用.在链路占用期间,所有路由经过这条BBP链路上任何一个端口的链路申请必将失败.

BBP_PVM能动消息通讯模式的引入使得BBP链路不再成为瓶颈.当一个消息到来时,消息接收处理机的当前进程被中断,同时一个message_handler被激活,迅速将消息取

出放于消息接收队列中，等待接收进程处理. BBP_PVM采用的能动消息工作模式不需要重写系统核心，它与当前的网络通讯协议完全共存. 图8是能动消息工作模式通讯过程图示.

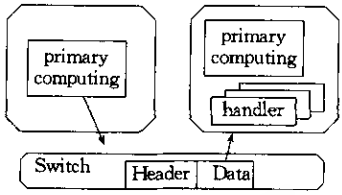

图8 消息传输机制

4.2 死锁预防机制

BBnet 监控每条 BBP 通讯链路时采用各链路平等争用和不可抢占策略. 一旦一条链路被占用，其它端口申请这条链路都将失败.

当两个端口同时申请一条链路时，双方有可能由于反复重新申请链路而陷于死锁. 这期间所有经过其中一个端口的其它链路申请也将失败. 为了避免这种死锁，BBP 规定链路申请裁决任一方一旦失败，必须立刻释放链路裁决权和在此之前已经占据的所有链路资源，在 BBP_PVM 全局流量控制机制的管理下重新申请 BBP 链路.

4.3 通讯设备共享

在多机系统中，软件的开销主要在消息传递的延迟上，尤其在于多次的数据拷贝、缓冲管理和可靠传输维护. BBP_PVM 的多进程设备共享机制提供了高效的网络接口，有效降低了软件开销. 在该机制的支持下，每对进程可以直接通过 BBnet 通讯，无需将这些数据交由一个专门的进程负责消息传递，减少了数据拷贝次数，减轻了数据缓冲管理的时间和空间负担. BBP_PVM 在采用通讯设备共享机制后，数据传输速度较不采用共享机制时提高1.5倍，链路建立时间下降50%.

5 结　论

BBP_PVM 的研制成功，为基于总线桥协议的异构可扩展并行计算机群 BBP_SPC 提供了一个有效的并行软件开发环境，大大提高了 BBP_SPC 系统的实用性和可操作性. 能动消息通讯模式等新技术的采用，大大降低了并行机群内通讯开销中软件成分所占的比例. 运行于 BBP_SPC 样机(Pentium PC 为处理器)系统上的 BBP_PVM 较以太网上的 PVM3.3 版在传输速率上快近3倍，通讯速率达1.2MBytes/s，链路平均建立时延约为1 900ms，较 PVM3.3版低约40%.

参考文献

1 Hwang Kai. Advanced computer architecture. Parallelism, Scalability, Programmability, McGraw-Hill, Inc. 1993.

2 金利杰，李未. 基于整机模块的高性能多计算机系统. 中国博士后首届学术大会论文集，1993.

3 金利杰，尹朝万等. 总线桥：概念、方法和实现. 机器人，1992，14(1).

4 金利杰. 可调耦合度多计算机系统的运行机理[博士论文]. 哈尔滨：哈尔滨工业大学，1992.

5 可扩展并行计算机群系统可行性论证报告(863-306-01-03-02)，1994-09-24.

6 Li Wei, Jin Lijie. The operational mechanism of a bus bridge network. Proceedings of APPT'95, Beijing, Sept. 1995. 26～27.

7 Jin Lijie, Li Wei. Design and analysis for the running mechanism and feature parameters of the bus bridge network. The 19th Australia Computer Science Conference, Melbourne, Australia, Jan. 30th～Feb. 2nd, 1996.

8 北京航空航天大学计算机系. 基于总线桥多机互联协议的可扩展并行计算机群. BBP_PVM 技术文档. 1996-02-12.

9 MPI: A message passing interface. Proceedings of Supercomputing 93, 1993.

THE CONSTRUCTURE OF A PVM DIALECT FOR A SCALABLE PARALLEL COMPUTER CLUSTER SYSTEM BASED ON BUS-BRIDGE-PROTOCOL

JIN Lijie ZHANG Jianjun LI Wei

(*Department of Computer Science Beijing University of Aeronautics and Astronautics Beijing* 100083)

Abstract The BBP_PVM is a dialect of PVM for a scalable parallel computer cluster system that based on Bus-Bridge-Protocol. The inter-machine communication protocol the BBP_PVM is called BBP_MPL, the message passing layer protocol of the bus bridge protocol. The BBP_PVM is compatible with the original PVM version 3.3.4 and above, but it has a more effective communication mechanism than the original version.

Key words Bus bridge protocol, reduced communication protocol, scalable parallel computer cluster, parallel virtual machine.

Class number TP302