一个基于 Windows 和 PVM 的 Beowulf 机群系统的设计与性能分析

祝永志[1] 魏榕晖[1] 赵本立[3]

(曲阜师范大学计算机科学学院 日照 276826)[1] (济宁职业技术学院 济宁 272000)[2]

摘 要 具有良好性价比的 Beowulf 机群系统在并行计算领域得到越来越广泛的应用。论文介绍了在 Windows 2000 Server 操作系统下基于 PVM3.4beta6 环境构造一个 Beowulf 机群系统的方法。利用一个并行算法实际测试了该 Beowulf 系统的并行计算加速比和并行效率。结果表明该 Beowulf 系统具有很高的并行计算效率和性价比。

关键词 Beowulf,并行计算,PVM,Windows

Designing and Evaluating the Performance of a Beowulf Cluster System Based on Windows and PVM

ZHU Yong-Zhi[1] WEI Rong-Hui[2] ZHAO Ben-Li[2]

(College of Computer Science, Qufu Normal University, Rizhao 273165)[1] (Jining Vocational Technology College, Jining 272000)[2]

Abstract The Beowulf system having gotten widely applications as it's good ration of price and quality. This paper introduces how to build a Beowulf system under Windows based on PVM3.4beta6. Tests using a parallel computing program measure the paralle efficiency of this system. Results show that this Beowulf system could run high-performance computing tasks.

Keywords Beowulf, Parallel computing, PVM, Windows

1 引言

随着现代科学技术的发展,大规模数据处理向人们提出了新的需求。并行计算机的出现为成功地解决这些问题开辟了一个可行的途径。同时,人们对于并行计算的需求空前提高,它的应用在朝着如下几点趋势发展:地理上高度分散,数据量庞大,信息流多维化,数据通信和管理的处理的工作量十分庞大等。

高性能并行计算机大致可分为五类:第一类是阵列机,开始于 20 世纪 60 年代后期,主要代表是 ILLIAC IV 阵列机;第二类是向量多处理机,开始于 70 年代,以 CRAY YMP-90、NEC SX-3 和 FUJITSU VP-2000 等为代表。在此阶段,向量多处理机的体系结构有了重大的发展,同时,向量识别和自动编译技术也有所突破。第三类是基于共享存储的多处理机系统,如 SGI Challenge 和 Sun Sparc Center 2000,同时,分布存储多计算机系统也开始出现。在此期间,并行设计技术方面有了进一步的提高和完善,体系结构也日趋成熟。当然,由于共享结构的限制,系统的规模不可能很大,因而系统的可扩展性受到了一定的限制;第四类是基于分布存储的大规模并行处理系统(MPP),开始于 80 年代末,90 年代初,典型的产品如 Intel Paragon、CM-5E、Cray T3D、IBM SP2 等;第五类则是基于 RISC 工作站或高档微机通过高速互连网络连接而构成的集群计算机系统。第二和第四类系统由于研制费用及成本较高等因素,其市场受到一定的限制。第三类系统由于受到结构的限制,处理器数目不可能做得很大。集群计算机系统由于具有投资风险小、可扩展性好、可继承现有软硬件资源和开发周期短、容易编程等突出特点,目前已很快成为并行处理的热点和主流。

随着微机性价比的提高和以太网(Ethernet)等局域网技术的成熟和硬件成本降低,以及消息传递标准和相应软件的发展,为用一组微机建立并行计算集群(常称为 Beowulf 系统)铺平了道路。

本文笔者构造了一个 Beowulf 并行计算系统,该系统由 16 台普通微机组成,它们由 100M b/s 高速交换式以太网相连接,每台微机上运行 Windows 2000 Server 操作系统,采用 PVM3.4.beta6 版本作为并行计算的支撑环境。最后,利用 PVM 的并行函数库,用 VC++ 6.0 编写了一个计算 Π 的 SPMD 程序,实际测试了该 Beowulf 系统的并行计算加速比和并行效率。

2 Beowulf 集群系统简介

集群系统是利用通用的高速网络将一组高性能工作站或高档 PC 机,按某种结构连接起来,在并行程序设计以及可视化人机交互集成开发环境的支持下,统一调度,协调处理,实现高效并行处理的系统。系统的资源管理及相互协作一般由操作系统之上的并行编程环境完成,因而如果设计合理,系统就可以屏蔽底层硬件(包括工作站和网络)的异构性,从而具有相当好的跨平台性能,同时也使得 NOW 的可扩展性更强。

Beowulf 系统将一些松散的计算资源——普通的 PC 机作成一个集群,实现以前只能由并行计算机完成的高性能计算。Beowulf 系统所具有的价格优势是传统的并行计算机所无法比拟的。

通常在 Beowulf 集群上运行的软件是 Linux 操作系统或 Windows 操作系统、并行虚处理机 PVM(Parallel Virtual Machine)和消息传递接口 MPI(Message Passing Interface)。一般由服务结点来控制整个集群。服务结点是集群的控制台和对外的网关。在规模较大的集群中可以有多个服务结点,如专门用集群中 Beowulf 的一个结点作为控制台或统计整个集群的运行状态。通常,除服务结点外,集群中的其他结点都是哑成员,即 Beowulf 它们不与外界交互。这些成员结点由

祝永志 硕士,副教授,主要研究方向:网络与分布式系统;**魏榕晖** 硕士研究生,主要研究方向:网络与分布式系统;**赵本立** 硕士,讲师,主要研究方向:网络技术。

服务结点来管理，执行服务结点分配的任务。Beowulf 集群中的成员结点以及内部连接是集群专用的。从这一点来看，Beowulf 更像是一台完整的机器，而不是一个由许多计算机组成的松散的群体。

3 Beowulf 集群系统的硬件和软件构造

目前在集群环境中应用较多的是消息传递模型。在消息传递模型中，各个并行执行的部分之间通过传递消息来交换信息、协调步伐、控制执行。消息传递通常是面向分布式内存的，但也适用于共享内存的并行机。消息传递为编程者提供了更灵活的控制手段和表达并行的方法，灵活性和控制手段的多样化是消息传递并行程序能提供高的执行效率的重要原因。消息传递模型一方面为编程者提供了灵活性，另一方面它也将各个并行执行部分之间复杂的信息交换和协调控制的任务交给了编程者，这在一定程度上增加了编程者的负担，这也是消息传递编程模型编程级别低的主要原因。

在当前所有的消息传递软件中，最流行的就是并行虚拟机 PVM(parallel virtual machine)和消息传递接口 MPI(message passing interface)。MPI 是一个显示的消息传递模式，在其中，任务通过发送消息进行相互通信。其最大的优点是高性能，点到点通信函数模型、可操作数据类型都比 PVM 丰富，群组通信的函数库也更大，但是不如 PVM 灵活。MPI 和 PVM 都提供了一套函数集，且各有所专。它们能运行在所有的并行平台上，包括 PVP、SMP、MPP(massively parallel processor)、工作站和 PC 组成的集群系统，并已经在 Windows 平台上实现，提供了对 C、Fortran 和 Java 语言的绑定。

为满足科研和研究生教学的需要，笔者构建了一套 Beowulf 并行计算系统，该系统采用由微机及高速以太网络组成的分布式、同构、对等集群结构形式，由 16 台微机组成。以太网由于结构简单、构造容易，可以低廉的价格获取较高的局域网数据传输性能，因而成为多数 Beowulf 并行计算系统的选择。

Beowulf 并行计算系统的微机操作系统多为：UNIX、Linux、WindowsNT/2000/XP。它们都具有很强网络支持功能和可靠性。结合实验室环境，本 Beowulf 并行计算系统的微机操作系统均为 Windows 2000 Server。安装好 Windows 2000 Server 后，对 Windows 2000 Server 进行网络配置。在所有的结点微机上安装 TCP/IP 网络协议，并将所有的结点微机设置为同一工作组，结点微机定义不同的网络名以便区分。再将结点微机上计划安装 MPI 软件和存储并行程序的硬盘分区或文件夹设置为共享。

3.1 PVM 环境下的开发工具

PVM 是美国 Oak Ridge 国家实验室开发的基于消息传递的并行软件系统，它基于 TCP/IP 协议将一个网络中各节点计算机虚拟成一个并行机使用，使高性能计算在廉价环境中得以实现。

在分布式存储的并行机中，消息传递(Message Passing)是一种被广泛使用的分布式计算程序结构模型。按照该模型，一个程序被分成几个子程序分别在各个结点上运行，并通过相互传递消息来保持整个程序的协调和同步。对起初的分布内存机，结点就是各 CPU。现在，随着计算机网络的发展，通过网络连接的计算机集群系统中已成为分布式计算的一种趋势。结点不仅可以是多 CPU 机的各个 CPU，还可以是一个网络上的各个 PC。程序员直接与网络打交道，编制与网络的接口程序；同时要负责网络上计算机间数据格式的转换，防止网络时延造成的不可靠性。由于程序分布在整个网络上执行，程序员还必须考虑一旦某个结点机失败后的容错问题。

消息传递通信系统是围绕消息(Message)概念来建立的，在并行程序设计环境中，一个消息即是由一个发送数据操作 Write 所产生的结果，其中数据为一些没有结构定义的字节流。每个消息由一个读取操作 Read 来消耗，如果消息字节数大于读取操作所需要的字节数，则多余字节将被丢掉，反之，则 Read 读取操作将读取所有此消息中的字节，同时返回消息字节数不够的指示。

(1)PVM 的主要特点：

1. 易于编程：PVM 支持多种并行计算模型，用户使用 PVM 提供的函数库可进行并行程序或分布式程序的设计工作，使用传统的 C 语言、C++ 语言和 Fortran 语言。

2. 消息通信：系统提供了一组通讯原语，可实现一个任务向其它多任务发消息，以及阻塞和无阻塞收发消息等功能，用户编程与网络接口分离。

3. 系统规模小：整个系统只占 3M 左右的空间，并且该软件系统是免费提供的。

4. 进程组：PVM 可以把一些进程组成一个进程组，一个进程可属于多个进程组，而且可以在执行时动态改变。

5. 支持异构性：计算机联网构成并行虚拟计算机系统且易于安装、配置。异构性体现在：一是并行虚拟计算机系统结构可以不同，支持向量机、多处理机、并行计算机、PC 机以及工作站。二是 PVM 允许虚拟机内的多个结点机用不同的网络相连，如 FDDI，Token RING 和 Ethernet 等。

6. 容错能力：当一个结点机出故障时，PVM 会自动将其从并行虚拟计算机系统中删除。

7. 多用户和多任务：多个用户可将系统配置成相互重叠的虚拟机。

3.2 系统组成

(一)每台微机的硬件配置为：

(1)主板：华硕 A7V8X；

(2)中央处理器：AMD Athron 2000+；

(3)内存：1GB；

(4)硬盘：40GB；

(二)网络设备

(1)交换机：16 口 100Mb/s；

(2)网卡：100Mbps 自适应网卡×16；

(3)超五类双绞网络线缆、接头：适量。

(三)系统和软件配置

每台微机上运行 Windows2000 server 操作系统，采用 PVM3.4beta6 作为并行计算的支撑环境。

4 性能测试与分析

加速比是并行计算的一个重要评测性能的概念。它指的是并行程序相比相同算法的串行程序所获得的性能提高的倍数。根据 Amdahl 定律，加速比：

$$S_p=\frac{Ws+Wp}{Ws+Wp/P} \quad (1)$$

其中 Ws 是应用程序中的串行分量，Wp 是可并行化部分，P 是并行系统中处理器数。可见对一定的计算负载，可将其分布在多个处理器上，通过增加处理器数以加快执行速度，从而达到了加速的目的。但当 $P\to\infty$ 时，$S_p=\frac{1}{f}$(其中 $f=Ws/W$，W 是计算负载)。这意味着随着处理器数目的无限增大，并行系统所能达到的加速之上限为 $1/f$。实际上并行加速不仅受限于程序的串行分量，而且也受并行程序运行时的额外开销影响。

(下转封四)

(上接第 279 页)

衡量并行系统性能另一个技术指标—并行效率。P 个结点的并行效率为：

$$Ep=\frac{Sp}{p} \quad (2)$$

理想状态下，加速比接近于 P，并行效率接近于 100%。

PVM 支持多种并行计算模型，常见的有 SPMD 模型和 Master/Slave 模型。Maste/Slave 模型（主从式），由一个 Master 负责任务的划分、分派和收集结果，由多个 Slave 负责子任务的计算，采用消息传递方式进行同步通讯。SPMD(独立进程式)模型，即等价模型，所有进程都是同一的，不同数据，各节点采用消息传递同步或异步方式通讯。

本文测试程序是用 C 语言编写的 PVM 计算 Π，它是一个 SPMD 程序。PVM 库(libpvm3.a)提供了很多并行程序设计所需的函数和过程。本文中使用了如下几类：

①进程的创建与控制：

Int pvm_mytid()/*进入 PVM，同时获得自己的 TID*/

Int pvm_prarent()/*返回任务的父进程的 TID*/

Int pvm_spawn(char * task, char * * argv, int flag, char * where, int ntask, int * tids)/*生成 PVMD 的子进程*/

Int pvm_exit()/*退出 PVM*/

②进程间的通讯：

Int pvm_inisend(int encoding)/*创建一个新的发送缓冲区，并置其为活动缓冲区*/

Int pvm_pkint(int * np, int nitem, int stride)/*数据发送前打包*/

Int pvm_send(int tid, int msgtag)/*将打包好的数据发送给其它的进程*/

Int pvm_recv(int tid, int masgtag)/*用于接收 tid 发来的消息 magtag，它是阻塞式接收函数，一直要等到消息到达后才返回*/

Int pvm_mcast(int tids, int nitem, int msgtag)/*把当前缓冲区的消息发送给 ntask 个任务(不含自己)*/

Int pvm_upkint(int * np, int nitem, int stride)/*将数据从当前缓冲区读出，在接收方，对数据的解包顺序应和打包顺序一样*/

③动态进程组：

Int pvm_joingroup(char * group)/*加入任务组*/

Int pvm_lvgroup(char * group)/*离开任务组*/

Int pvm_barrier(char * group, int count)/*调用任务等待组中共有 count 个任务调用了 pvm_barrier()*/

Int pvm_reduce(void(*func), void * data, int nitem, int datatype, int msgtag, char * group, int root)/*用于在指定的任务组的全体成员进行全局性运算*/

从实验中看出，随着问题规模的增大，CPU 利用率先递减后递增，这是因为当问题规模较小时，程序的通讯量并不大，与用户时间相比，系统时间占了大比例。当问题规模增大后，用户需要的计算量不断增大，用户时间上升的较快，而系统时间较慢。另一方面，可见，只有当计算量超出通讯量时，并行程序才能发挥它的作用，并且，计算通讯量越大，并行程序的性能越高，并行效果越好。问题规模较小采用并行方法求解有时适得其反，尤其在分布环境下。

结束语 由于 Beowulf 系统不但可以获得较高的计算能力，同时具有极为优越的性价比，因此在中小科研单位和高校利用 Beowulf 集群系统来进行并行计算是一种趋势。PVM 最初构建在 UNIX OS 上，后来出现了基于 Windows 版本。在我国使用 Windows 操作系统的用户很多，但构建实用、高效的基于 Windows 和 PVM 的 Beowulf 并行计算系统和相关资料较少，本文所构建的系统经过实际测试取得了很高的并行效率且有很高的性能价格比，为 Beowulf 系统的成功创建提供很好的例证和经验。

参考文献

1 彭雷，朱永芬，戴光明. PVM 下矩阵相乘并行算法的研究与实现[J]. 微机发展，2004，14(8)：49～51

2 陈星，黄卡玛. 构建基于 Windows 和 MPI 的 Beowulf 并行计算系统[J]. 计算机工程与应用，2003，39(11)：59～61

3 陈国良. 并行计算——结构·算法·编程[M]. 北京：高等教育出版社，2003. 364～368

4 李贵明，俞国扬，罗家融. 基于 Linux 的 Beowulf 集群的实现[J]. 计算机工程，2003，29

5 尚月强. Windows2000 下基于 PVM 的并行计算实践研究. 计算机系统应用，2005，4：67～70

计算机科学

(1974 年 1 月创刊)

第 33 卷第 6 期 (月刊)

2006 年 6 月 25 日出版

国际标准连续出版物号 ISSN 1002－137X

国内统一连续物出版号 CN50—1075/TP

定价：30.00 元　国外定价：5 美元

邮发代号：78—68

发行范围：国内外公开

主管单位：国家科学技术部

主办单位：国家科技部西南信息中心

编辑出版：《计算机科学》杂志社

重庆市渝中区胜利路 132 号　邮政编码：400013

电话：(023) 63500828　E-mail：jsjkx@swic.ac.cn

网址：www.jsjkx.com

社　长：牟炳林

总　编：彭　丹

主　编：朱宗元

主编助理：徐书令

印 刷 者：重庆科情印务有限公司

总发行处：重庆市邮政局

订 购 处：全国各地邮政局

国外总发行：中国国际图书贸易总公司（北京 399 信箱）

国外代号：6210—MO