并行程序设计 程序设计 PVM

计算机科学 1996 Vol. 23 No. 1

32-35

基于 PVM 的分布式并行程序设计方法及其应用

党华锐 電帮军 郑守淇

(西安交通大学计算机科学与工程系 西安 710049)

TP311.11

摘 要 The history, principle of implementation and usage of PVM (Parallel Virtual Machine) are described in briefly. Meanwhile, the paradigm of PVM and its application in distributed computation are introduced.

关键词 Network, Parallel, Distributed computation.

近年来,基于消息传递的并行处理方式越来越受到重视。PVM(Parallel Virture Machine)正是这样一种软件系统,它允许用户将通过网络相互连接的众多异种机看成是一个单一的虚拟并行计算机系统,这些异种机可以是多处理机系统,也可以是工作站或 PC 机,而异种机间的消息传递、数据转换和任务分配等均由 PVM 以透明的方式自动完成。所以,对应用程序来讲 PVM 是一个基于消息传递的分布式并行计算环境。

PVM 的研制始于 1989 年夏,主要参与的机构有美国 Oak Ridge 国家实验室,田纳西大学、Emorg 大学、卡内基梅隆大学等,同时,本项目还得到了美国能源部、国家自然科学基金会和田纳西州政府的资助。

作为免费交流的软件,PVM 已出了三版,目前的最新版本为 PVM3.3.7 测试版。同时,由于 IBM、DEC 等大公司已决定将 PVM 移植到自己的新机器上,所以 PVM 的商品化工作也在进行之中。

1. PVM 的实现原理

从软件结构上讲,PVM 由两大部分组成:(1)后台常驻进程 pvmd:它驻留在组成虚拟机的所有机器上,相互之间以消息传递机制互相联系;(2)接口函数库 libpvm.a:主要提供了与用户的接口函数,使用户程序可方便地利用 PVM 提供的机制和资源。该函数库主要包含了用于消息传递,启动进程,调度任务及修改虚拟机配置等的函数。

另外,PVM 还有一个供用户使用的控制台,在控制台中,用户可以以命令的方式查询并在适当的时候改动虚拟机的配置和方式,并且,PVM 每次启动时总是先判断 pvmd 的存在(以决定是否装入),然后才启动控制台。

PVM 是利用 UDP/TCP 在任务间传递消息以实现其同步的,常用的同步方法有三种:信号机制(signals),阀调用(barriers)和发送通知(notification)。

以下将先对 pvmd, libpvm.a 分别做一简单介绍,然后阐述两者之间的接口。

1.1 后台常驻进程 pvmd

(1)启动过程:首先,pvmd 使所在主机的输入、输出工作于正常状态,获取自身所处的环境信息(包括用户标识、主机端口号、用户口令、pvm 系统所在主目录"PVM_ROOT"及主机的体系代号),并建立注册文件 log_fd。为了支持多处理机,一并对多处理机进行初始化。环境确定后,pvmd 处理命令行参数并据此对 pvmd 自身的"主机表"结构(主 pvmd 的"主机表"包括虚拟机中的所有主机情况,而各从属 pvmd 的"主机表"则仅包括主 pvmd 所在

党华锐 讲师,博士生:雷邦军 硕士生:郑守淇 教授,博士生导师。主要研究方向为:新型计算机结构、并行程序设计、人工智能等。

主机和其自身)。同时,pvmd 一并对自身的各种队列和结构进行初始化并设置各种信号的处理方式。最后,若为主 pvmd,则向虚拟机中其它主机发信号以告之启动从 pvmd,然后记录初始时间参数后转入循环工作状态,对整个虚拟机的工作进行控制;若为从属 pvmd,则直接在记录初始时间参数后转入循环工作状态,以负责本 pvmd 下的各并行任务的运行及与其它 pvmd 之间的通讯。

(2)内部工作机制:虚拟机中每个主机上都有一个 pvmd,其中启动整个 pvm 系统的主机上的 pvmd 称为主 pvmd,其它主机上的 pvmd 称为从属 pvmd。PVM 假定虚拟机中各主机之间通过 IP 互联。各 pvmd 之间通过 UDP 接口通讯,pvmd 和本地任务之间的直接通讯也建立在 UDP/TCP 之上,而在多处理机中两节点之间则使用本地通讯函数(使用共享存储或消息硬件机制)。每个 pvmd 在循环工作状态即正常后台控制时主要做四项工作:

A. 任务管理:每个 pvmd 维护在该主机运行的所有任务并为每个任务记录上下文;B. 等待结构:当 pvmd 要求远程服务而等待时,会事先将所有有关环境数据保存在一个"等待上下文"(waitc)中,以便在接收到回复后恢复现场继续工作;C. 机器重组:当虚拟机中新加入一个主机时,主 pvmd 启动自身的一个副本(称为影子 pvmd)。该影子 pvmd 用 rsh()或 exec()启动每个新 pvmd,并在完成后将结果通过 pvmd-pvmd 消息通道传给主 pvmd;D. 错误检测:pvmd 将所导致应用程序永挂起的错误转换成可检测事件,如消息。为了检测主机,每个 pvmd 掌握一个计时器,定期向其它 pvmd 发送消息。若在规定时间内未收到回复,则认为那个主机出错,并将其从虚拟机中去除。另:若设置了 STATISTICS,pvmd 还要进行一些统计工作。

1.2 接口函数库 libpvm.a

主要包括以下几类函数,可供用户在 C 或 Fortran 程序中直接调用以使用 PVM 系统:进程控制;各种信息获取;虚拟机动态配置;信号发送;错误信息处理;消息传递。

1.3 libpvm 与 pvmd 的连接

首先,libpvm 读文件/tmp/pvm.{uid},它是在 pvmd 启动时由 pvmd 建立的,其中包含了 pvmd 将接受来自任务的连接接口地址("ip_addr:port")。libpvm 将一个套接字连到这个地址上并向 pvmd 发一消息请求进入。当连接建立后,pvmd 和 libpvm 必须互相证实身份。pvmd 和任务各创建一个仅被它们自己的用户拥有和可写的文件,它们互相交换文件并试图向对方写文件,然后互相检查自己的文件,以防止非法用户闯入。当一个任务"重新链接"至 pvmd,它必须给 pvmd 提供它 pid,也就是说,一个任务在 spawn 操作中以 fork()返回,pvmd 会为任务创建一个上下文并填入该 pid 中。当在 debugger 下运行一个任务,该任务将成为 debugger 的子进程并拥有一个 pid。在 exec 一个任务前,pvmd 将环境变量 PVMEPID 设置为该任务的 pid,PVMEPID 设置后,libpvm 用它自己的值作为 pid 以向 pvmd 证实自己,接着发送实际的 pid。

2. 并行虚拟机(VM)的建立[1]

2.1 PVM 的获取

目前,PVM 作为非商品化的测试软件提供于 INTERNET 上,用户可从 INTERNET 上的任一结点利用 INTERNET 提供的如下几种方法中的任何一种来获取 PVM。(1)Anonymous ftp:地址为 netlib2.cs.utk.edu。(2)World Wide Web:www http://www.netlib.orl/pvm3/index.html。(3)E-mail:地址为 netlib@ornl.gov。

2.2 PVM 的安装

PVM 是以 UNIX 操作系统为其运行平台,且要求安装在每一个 HOST 上,现以 SUN 工作站为例说明

(1)解开压缩文件。此操作将创建 PVM 的各级目录,并展开所有的 PVM 文件。

(2)选择 SHELL。在编译、链接并生成可执行文件前(即做 make 前)要设置相应的环境变量,以下工作均对 csh 而言,若用非 csh,如用 B shell,则要在.profile 中用不同的命令来设置(请参考/home/pvm3/Readme)。

(3)设置环境变量。

setenv PVM_ROOT/home/pvm3

setenv PVM_ARCH SUN4

此后变量 $PVM_ROOT 就指/home/pvm3,

PVM_ARCH 就指 SUN4。

(4)编译。

```
cd /home/pvm3
make
```

编译后 Daemon 为/home/pvm3/lib/pvmd；PVM 的 Console 为/home/pvm3/lib/SUN4/pvm。

(5)HOST 上例子的编译为链接。PVM 本身带有一些示范的例子，这些例子在运行前必须先编译和链接，具体做法为：

```
cd /home/pvm3/examples
$PVM_ROOT/lib/aimk
```

生成的可执行程序在/home/pvm3/bin/SUN4 目录下。

2.3 VM 的构成方式[4]

PVM HOST 间的消息传递是建立在 UNIX 的 Remote Shell 之上的，所以在运行 PVM 之前应先检验一下 UNIX 的网络结点是否能利用其进行通信，以及 PVM Daemon 是否能正常启动。要使用 PVM 进行并行计算则首先要启动所有 HOST 上的 Daemon，此工作只需在一个 HOST 上做即可。有两种启动方式：命令行(pvm)和系统调用(pvm_addhost())。

3. PVM 的程序设计风范

应用程序可在三个不同的层次上使用 PVM：

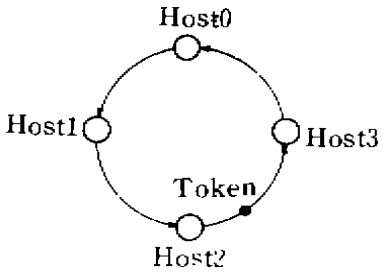
高层：这是一种透明的使用方式，应用程序无需考虑具体的 PVM 系统结构，由 PVM 自动选择合适的 HOSTS 来进行分布计算。

中层：与结构有关，用户可以指定某一种(类)结构的机器来运行特定的任务，这相当于"组(GROUP)"的概念。

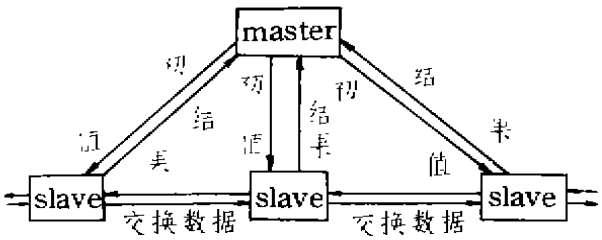
低层：与具体 HOST 有关，用户可以直接指明在哪一台 HOST 上运行。

PVM 的任务，相当于 UNIX 的进程，它们可以并发执行。PVM 对其并发方式不加限定，所以，PVM 可以支持最一般的 MIMD 并行计算。例如，常用的并发方式有：

(1)SPMD，此时所有的任务彼此平等、不分主次。如例子 spmd.c，在所有的 HOSTS 上都驻留同一段可执行程序 spmd，其作用是依次传递令牌，构成模拟的令牌环，其相互配合的方式可用下图表示：



(2)MASTER/SLAVE，一群 SLAVE 任务为一个或几个 MASTER 服务。如例子 master1.c/slave1.c，主程序 master1 控制多个从程序 slave1 进行计算，在计算的过程中，从程序可以相互交换数据，最后，众多的 slave1 将各自的计算结果传给 master1，由它进行汇总，其作用方式如下图所示：



在使用 PVM 时，应用程序需要考虑以下几方面的问题：(1)任务的粒度。在 PVM 中，任务的粒度主要以待传递信息的字节数或任务中所含的浮点运算量来衡量，一般的情况是，粒度越大，SPEEDUP 就越大，并行程度就越低。(2)信息包的大小，包不能太大，否则会加重系统的负载，而小的包由于能和计算重叠进行，则系统的负载较轻。(3)适当选择"功能并行(FUNCTION PARALLEL)"和"数据并行(DATA PARALLEL)"。在 PVM 中，功能并行指在 PVM 的各 HOST 执行不同的任务；而数据并行指将数据分解并分布于各 HOST 之上，各 HOST 做相同(相似)的工作(这种方式更适用于多处理机系统，因为只需由一段可执行程序即可)。

PVM 消息传递的一般格式

[发送方]	[接收方]
(1)初始化缓冲区 pvm_initsend()或 pvm_mkbuf()	(1)初始化缓冲区，接收消息 (2)拆包
(2)打包 pvm_pk*()	
(3)发送 pvm_send()或 pvm_mcast()	

总之，消息的发送和接收都是围绕着活动缓冲区进行的。

4. 在 PVM 之上开发应用程序的方法

4.1 应用程序的框架

```
#include "pvm3.h"/*调用 PVM 的函数库*/
main()
{
...
pvm_mytid();/*注册进入 PVM*/
...
pvm_exit();/*注消退出 PVM*/
}
```

4.2 编译与运行应用程序

用 UNIX 的编辑命令，PVM 的程序风范，编辑用户自己的程序；

用 PVM 的 aimk 编译、链接上述程序(但得对 Makefile.aimk 做适当的修改)；

启动 PVM，构成 VM；

运行之。

更简便的办法是用户用 UNIX 的 CC 来生成可执行的程序。例如，自己编一段名为 my.c 的程序，其功能是显示自己的任务号，源程序如下：

```
#include(stdio.h)
#include"pvmk3.h"
main()
{
    int mytid;
    mytid=pvm_mytid();
    printf("This is my task id %d\n",mytid);
    pvm_exit();
}
```

然后，用下列命令编译、联结，以 SUN 为例。

cc-O-I/home/pvm3/include-o my my.c-L/home/pvm3/lib/SUN4-lpvm3 生成的可执行程序为 my，它可在 UNIX 环境下直接运行。

5. PVM 的应用及研究趋势

PVM 已广泛用于科学计算[1,3]，如材料分析、天气预报、声波分析、图象处理等。另外，PVM 作为一种教学平台，广泛用于分布式并行程序设计的教学和研究中，如我们在其上开发的并行 Lisp 就是一例。

目前，PVM 的主要研究趋势为：(1)共享环境下 PVM 的系统优化；(2)PVM 执行的可视化；(3)分布环境下虚拟共享的实现；(4)性能分析；(5)集成环境(能支持各种并行结构)。

参考文献

[1] G. A. Geist et al., PVM: Parallel Virture Machine-A User's Guide and Tuterial for Networked Parallel Computating, MIT Press 1994

[2] V. S. Sunderam et al., The PVM Concurrent Computing System: Evolution, Experience, and Trends. Parallel Computing, Vol. 20, 1994

[3] R. E. Ewing et al., Distributed Computation of Wave Propagation Models Using PVM. ACM Proc. Of Super Computation, 1993

[4] G. A. Geist, et al., PVM User's Guide And Reference Manual, Oak Ridge National Lab. 1993

第四届中国人工智能联合学术会议(CJCAI-96)征文通知

《第四届中国人工智能联合学术会议 CJCAI-96》定于 1996 年 10 月在北京中国人民解放军国防大学召开，会议将安排特邀专题报告、论文报告及专题讨论。现将征文有关事项通知如下：

一、征文内容：面向人工智能及其应用的系统结构、语言；认知模型；人工智能与教育；知识工程与专家系统；人工神经网络；机器学习；知识获取；分布式人工智能；自动推理；模式识别；虚拟现实与人工生命系统；自然语言处理，机器翻译；机器人；多媒体、超媒体与人工智能；知识表示；遗传算法；人工智能集成技术(模糊专家系统、模糊神经网络等)；知识发现；智能控制；人工智能应用；人工智能理论基础；其它有关人工智能的论文。

二、论文要求：1)有创见、有实质内容并未发表过的；2)行文流畅、叙述清楚，字数不超过 8000 字；3)写清题目、作者姓名、工作单位及通讯地址(包括邮政编码)；4)中英文摘要：100～200 字(英文摘要另附)；5)关键词：不超过 4 个；6)来稿一式 2 份邮寄至承办单位联系人。请作者自留底稿，会议不退原稿。

三、截稿日期：1996 年 4 月 30 日(以邮戳日期为准)录用通知发出日期：1996 年 5 月 31 日

承办单位：中国人民解放军国防大学

联系人：周萍　邮编：100091　地址：北京 981 信箱电教中心　电话：(010)66769537