MPP 虚拟多机系统*

瞿国平 阳爱民 黎升洪 杨清 曾斌

(国防科技大学计算机系 长沙 410073)

摘 要 MPP 虚拟多机系统是一个特殊的 PVM 系统，本文阐述了 MPP 虚拟多机系统的系统结构和实现技术，并对 MPP 虚拟机的性能进行了初步测试和分析。

关键词 虚拟计算机，靶机，仿真

分类号 TP302.1

MPP Virtual Computer System

Qu Guoping Yang Haimin Li Shenghong Yan Jim Zen Bin

(Department of Comoputer Science, NUDT Changsha, 410073)

Abstract MPP virtual multiple computer system is a particular pvm system. This paper discusses the architecture and techniques of MPP Virtual Computer system, and a simple test and analysis of the MPP Virtual Conputer system are given.

Key words virtual computer, destination machine, simulation

MPP 虚似多机系统提供 MPP 多机系统调试环境。它能帮助用户进行多机程序正确性调试，还能进行用户程序优化、负载平衡分析、PVM 使用分析和存贮访问分析。在 MPP 多机系统研制阶段，它可作为系统程序的调试工具，MPP 虚拟多机是 MPP 多机系统的一个很好的辅助和补充。

MPP 虚拟多机系统的逻辑结构和目标多机的逻辑结构是一致的，它由一个宿主机，或多个同构型宿主机，或许多异构型宿主机构成。因此，实际上它就是一个特殊的 PVM 系统，因为它有 MPP 多机系统的一些特点，在用于计算方面，它比通常的 PVM 系统和目前广为流行的 JAVA 虚拟机有着明显的优点，如使用 MPP 虚拟多机的指令全局访问，用户用一条指令就可访问到另一个节点的信息等等。这种虚拟多机系统的构成和实现是一个很复杂的工作，它不仅要将各种类型的宿主机用软件改造成目标靶机的一个或多个工作单元 (PE)，而且这些 PE 之间要进行相互访问、相互通讯、同步工作、集中式和分布式控制。本文主要是阐述这方面的内容。

1 MPP 虚拟多机系统结构

有多种类型的多机系统，但就内存访问而言，主要有共享、分布式、分布式共享这几种类型的多机系统。共享内存多种系统中的所有处理单元 (PE) 都访问同一个公共存贮器，因此 PE 之间信息交换比较容易。国产的 YH-2、Crayc90、Crayc92 等属于这种类型的多机系统。分布式内存多机系统是由包含处理单元(PE) 和局部存贮器的节点构成，PE 之间的通讯主要通过 Message passing 来完成。Cm-5，paragon 等属于这种类型的多机系统。分布式共享内存多机系统同样由节点来构成，但每个节点的存贮器分为局部和全局两部分，全局存贮器可为多机系统中的所有 PE 访问，这种机器 PE 之间不仅可以通过 message passing 方式传送信息，还可以通过块传输方式和指令的全局访问来传送信息，KSR1/2 和 CrayT3D 等就是这种类型的机器，显然分布式共享内存的多机系统有着很明显的优点，因此，也是目前最为流行的多机系统，MPP 多机系统属于这种类型的机器。

* 1998 年 6 月 25 修订
第一作者：瞿国平，男，1946 年生，研究员

MPP 多机系统一般含有: 一组处理单元(PE)、存贮子系统、分布式存贮互连网、一组I/O Gateway 子系统等等。

MPP 虚拟多机系统主要由许多的虚 PE 来构成, 这些虚PE 分布于网络的各个节点上, 每个节点上可以有多个虚 PE。因此, MPP 虚拟多机系统可以在一个局域网上构成, 也可以在多个局域网, 甚至广域网上构成。图 1 是一个 MPP 虚拟多机系统拓扑结构图。

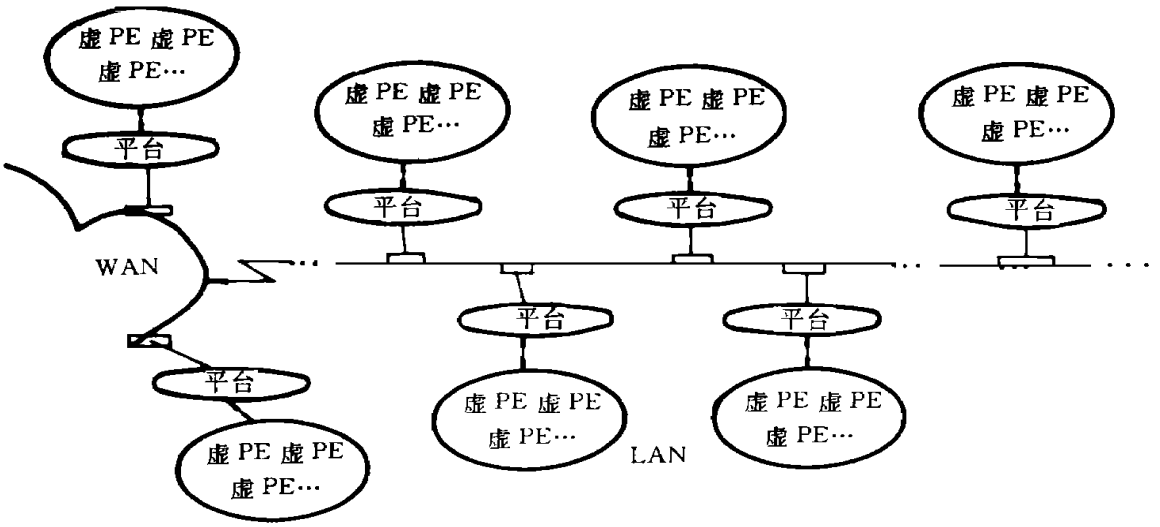

图 1 MPP 虚拟多机系统拓扑结构图

MPP 虚拟机的系统结构由两部分组成: 模拟靶机的部分和虚拟机辅助部分。图 2 是MPP 虚拟机的系统结构图。

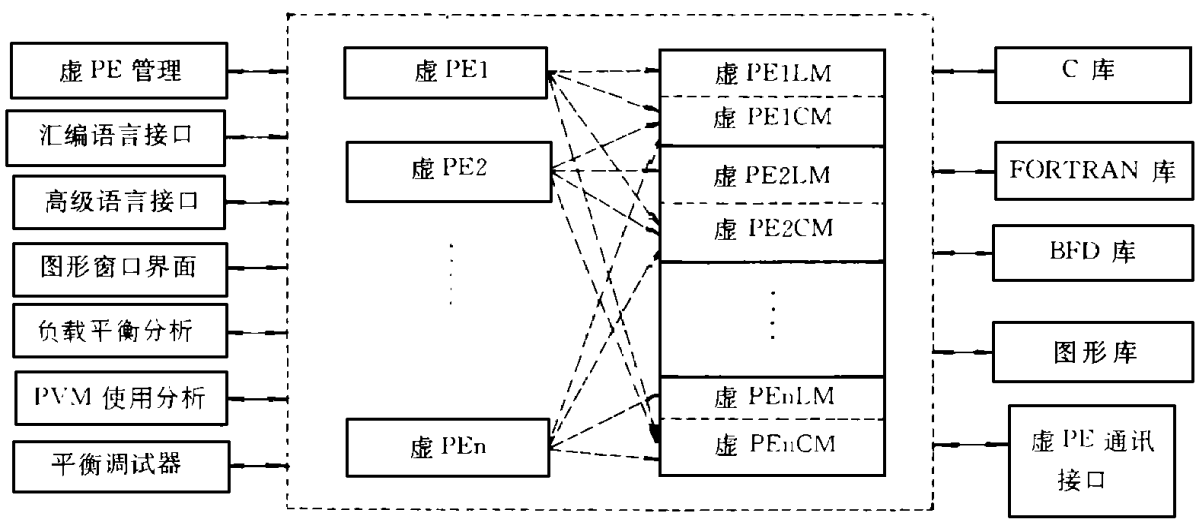

图 2 MPP 虚拟多机系统和系统结构图

MPP 虚拟多机系统是建立在虚 PE 基础上的, 每个网络节点上的虚 PE 产生器产生出所需的虚 PE, 根据宿主机所能提供的存贮空间平均地分配给所产生的各虚 PE。所有网络节点上的所有虚 PE 构成了MPP 虚拟多机系统的虚PE。每个虚PE 除能访问自己的虚局部存贮器处, 还能通过指令的全局访问、message passing 和块传输方式访问其它虚 PE 的虚全局存贮器, 这些虚 PE 在虚控制节点 (Host) 的控制下, 完成用户目标程序加载, 目标程序的模拟执行, 并在需要时, 可调用虚拟多机的辅助系统, 如平行调试子系统、优化用户程序子系统、反汇编器、反编译器和信息统计工具等来调试和优化用户程序。在虚拟机上运行的目标代码, 还可以通过虚拟多机的库接口访问到宿主机上各种库程序。

2 虚 PE 产生器

MPP 虚拟多机的一组虚 PE 是由一个节点或多个节点上的虚 PE 构成的, 每个节点上的虚 PE 是由虚PE 产生器产生的, 虚 PE 产生器是在单个虚PE 的基础上产生出多个虚PE。单个虚PE 是用模拟的方法模拟目标 MPP 多机系统的一个 PE。单个PE 模拟比单机的模拟复杂得多。单个 PE 不仅含有单机的一些特点, 更主要的它有很多多机的特性, 正是这些多机的特性使单个 PE 可以成为多机的一部分。

虚 PE 产生器就是在这样一个具有多机特性的虚 PE 的基础上, 用创建新进程, 让这虚 PE 在其上运行的方法产生出所需数目的运行着的虚 PE 进程。一个虚 PE 进程就代表 MPP 虚拟多机的一个虚PE。从理论上讲, 用unix 的fork 或sproc 都能产生出多个虚PE, 但实际上虚PE 产生器用 sproc 产生多个虚 PE 来构成虚拟多机比较好。这是因为每个虚PE 都有一个任何虚 PE 都能访问到的很大的公共虚存贮器, 如果用 fork 产生虚 PE 则很难用存贮共享的办法使每个 PE 访问到在其它进程中的别的虚PE 的公共存贮器。这是因为用 Fork 产生的各进程间的存贮共享是通过系统空间来实现的, 系统空间中能提供共享的区域是很有限的, 一般只有几 K 字节。而用 sproc 产生的进程间的通信通过公用语句在用户区就可实现共享, 因此, 可以共享的区域是很大的, 一般可以到几十兆字节。这几十兆共享的区域就可用为虚 PE 的全局存贮器, 一个虚 PE 对另一个虚PE 的访问和通讯, 就象访问自己的一个公用区那么容易。当然, 可以用 Socket 来解决用 Fork 所产生的虚 PE 间的相互访问, 但这将增加访存开销。因此, 在一个节点上还是用 sproc 产生多个虚 PE 为好。

3 虚 PE 间相互访问和通讯

虚 PE 间的访问和通讯是通过指令全局访问、message passing (mp) 和 BLT 块传输来实现的。不同节点虚PE 间的相互访问和通讯是利用报文, 通过 socket 发送和接收报文来实现的。因此确定好报文的格式和报文的种类, 这是虚 PE 间进行有效访问和通讯的根本保证。可以根据靶机的具体情况确定报文的格式和种类。图 3 是我们使用的虚 PE 间访问和通讯的报文格式。

其中类型码表示了报文的种类, 我们确定了如下一些报文码:

1——存数报文　　2——取数报文　　3——取数回答报文

4——mp 报文　　5——MP 回答报文　　6——BLT 报文

7——命令报文

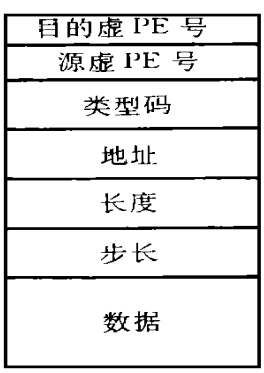

图 3　数据报文格式

对于不同类型的报文, 地址、长度、步长、数据这几个域各有不同的含义。虚 PE 根据不同的访问和通讯组织成不同的报文放入输出队列交发送进程, 发送进程从输出队列取出报文组织成网上发送报文, 使用 socket 发送报文。接收进程将接收的报文根据不同的报文码提交给虚 PE 或专门的处理进程处理。一个节点上有一个接收进程和一个发送进程为节点上所有虚 PE 服务。

(1) 指令全局访问模拟

a、存数访问

虚 PE 在模拟执行存数指令时, 如检测到的是对于其它节点上虚PE 公共存贮器的存贮访问, 则组织存数报文放入输出队列, 报文中包括所存数据地址, 长度和所要存的数据。在将存数报文放入输出队列后, 这条存数指令的模拟完成, 该虚 PE 继续下面指令的模拟执行。

b、取数访问

虚 PE 在检测到的是对于其它节点的虚 PE 进行取数访问时, 就组织取数报文, 并将它放入输出队列, 然后该虚PE 处于睡等所取数据到来。发送进程从输出队列取出报文, 并发送该报文到目的虚PE, 目的虚 PE 的接收进程接收到该取数报文后, 根据目的 PE 号就从相应虚 PE 的全局存贮器取到数据, 组织成取数回答报文。放入发送队列交发送进程发送给本节点, 当本节点的接收进程检测到接收到的是取数回答报文时, 就激活处于睡等状态下的虚 PE 并将数据交给该虚 PE, 该虚 PE 醒后, 使用所取

到的数据继续取数指令的模拟。

(2) Message passing (简称 MP) 模拟

为了模拟 MP, 首先必须定义与 MP 有关的硬件设备, 如消息缓冲池、缓冲池入口地址、MP 输入地址、MP 启动地址等等。除缓冲池外, 其余的都占用地址空间地址, 因此在存贮指令模拟中, 必须检测这些特殊的地址。当检测到存贮的目的地址是缓冲池入口地址时, 就将存贮的数据存到缓冲池。当检测到 MP 启动地址的启动位置起时, 就进行 MP 消息传输模拟: 将缓冲池中的信息组织 MP 报文, 放入输出队列, 由发送进程发送该报文。当目的虚 PE 的接收进程接收到这个 MP 报文时, 就根据报文中的目的虚 PE 号和本 PE 的 MP 输入地址存放 MP 报文的信息, 如需要回答, 则组织回答报文放入输出队列、交发送进程发送。

(3) BLT 模拟

因为 BLT 是大批量数据传输, 因此在模拟时, 不能用 MP 同样的方法。模拟中, 在 BLT 传输启动后, 由相应虚 PE 的 BLT 处理程序组织并放入输出队列的 BLT 报文, 不包含 BLT 数据, 只包含源和目的虚 PE 号, BLT 地址、长度、步长, 发送进程从输出队列取到 BLT 报文后, 根据报文中的信息, 从源 PE 的相应地址取到 BLT 数据, 并压缩数据, 组织报文, 并发送报文, 在发送后, 如发现还没有完成 BLT 数据的传输, 则继续组织报文发送数据, 一直到 BLT 数据传送完成。

在目的虚 PE 节点接收进程接收到 BLT 报文后, 就将 BLT 报文交给一个处理 BLT 报文的进程处理, 而接收进程立即使用新的缓冲区将接收口打开。BLT 处理进程根据报文中的目的虚 PE 号、地址、步长, 还原数据并存放到目的虚 PE 的内存中。

根据 (1) (2) (3) 我们可用图 4 表示各节点用于访问和通讯各实体之间的关系。

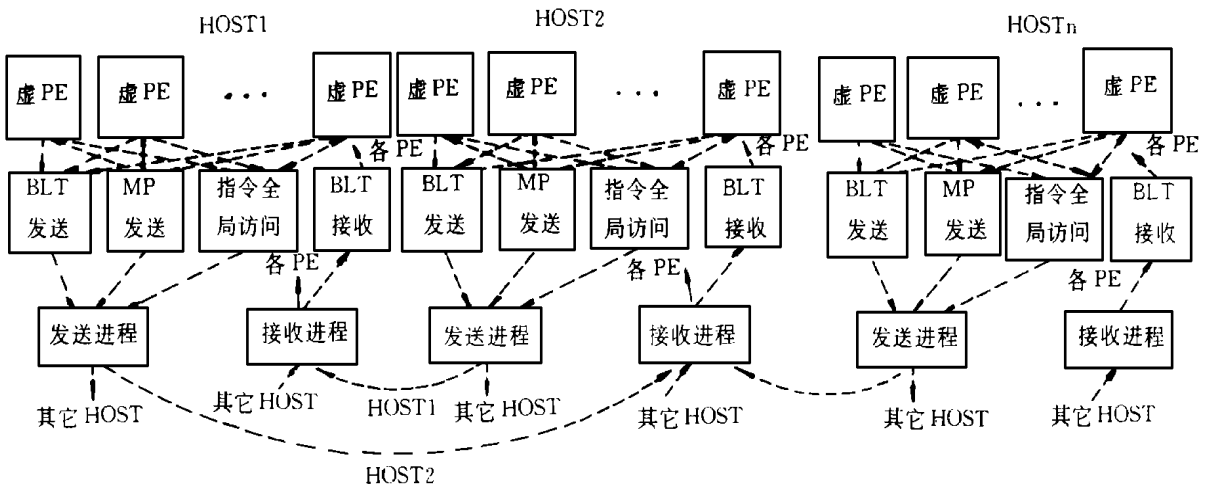

图 4 MPP 虚拟多机系统各实体之间关系图

4 MPP 虚拟机性能初步测试和分析

对 MPP 虚拟机的性能, 在最简单的情况下进行了测试, 即在宿主机都是 SGI, 而每个宿主机上只有一个虚 PE 的情况。测试的目的是: (1) 确定一个节点上 MPP 虚拟机使用宿主机的效率; (2) 确定不同节点上的虚 PE 协同完成一作业时使用网上结点的效率, 从而能大致估计出一个中等规模局域网上 MPP 虚拟机的运算速度。

为了确定 MPP 虚拟机使用宿主机的效率, 我们选用了三个典型的串行测试题, 分别在宿主机和 MPP 虚拟机上运行 (在 MPP 虚拟机上运行的目标代码在类似机靶机的 P8000 上编译)。表 1 是它们运行的结果。

表 1

测试题名称	宿主机运行时间（秒）	MPP 虚拟机运行时间（秒）	使用效率
Linpack	94．2	129．55	73．1%
WheStone	1618．33	2345．4	69．5%
Spec95	523．4	734．1	71．4%

由表 1 可以看出，MPP 虚拟机使用宿主机的效率大概在 70% 左右。

为了测试虚 PE 分布于不同网络节点上的 MPP 虚拟机使用网络节点的情况和虚 PE 件相互访问的开销，我们选用了两个典型的平行测试程序，分别在网络的 8 个节点上和以这 8 个节点为宿主机的 MPP 虚拟机上运行，宿主机在 PVM 方式下，使用 message passing 来协同完成测试题。MPP 虚拟机分别使用 message passing、BLT 和指令的全局访问来运行测试题。运行结果列于表 2。

表 2

测试题名	宿主机运行时间（秒）	MPP 虚拟机（秒）		
		MP 方式（使用率）	BLT（使用率）	指令全局访问（使用率）
NAS	22846	25697（63%）	33351（68．5%）	32966（69．3%）
NPB	52346	80905（64．7%）	33946（69．3%）	32590（70．1%）

显然，在 MPP 虚拟机上，使用 BLT 或指令全局访问所花代价比使用 MP 代价要小些。

在一个有几十台或上百台如 SGI、SUN 这样的机器和数百台 PC 机的局域网上，将这些机器都作为 MPP 虚拟机的宿主机就可协同完成大型的运算题目，而能获得每秒数几十亿的运算速度。

MPP 虚拟机的研制是一项复杂的工程，在现有的研究基础上，我们将不断地对系统进行进一步的完善。

参考文献

1 Qu Guoping. Function iustruction Simulation. CG&CAD 89′Proc. 46-62

2 Qu Guoping. An assembly-level Simulation System. Advances in Modelling&Simulation, 21 (390): 53-64

3 Qu Guoping. A distributed System of the Instruction Simulation. Advances in Modelling&Simulation (AMSE), 25 (3): 31～48

4 Qu Guoping. The disassemble-level Simulation System. Advances in Modelling&Simulation (AMSE), 25 (3): pp11～30

5 Qu Guopin. A Research on the Instruction Level Simulation of the Digital System. Advances in Modelling&Simulation, 21 (390): 35～36