实现 SMP机群虚拟化的方法

彭近兵 龙 翔 肖利民

(北京航空航天大学 计算机学院, 北京 100191)

摘 要: 机群是一类重要的并行计算机体系结构,成本低廉,可扩展性强,但编程复杂,管理难度大.利用虚拟化技术可以将机群抽象成具有共享内存特征的虚拟机,支持共享内存的编程模式,从而克服机群的不足.当前,在该领域有少量探索性的研究成果,但都不能支持对称多处理机(SMP, Symmetric Multiprocessor)机群虚拟化.本文提出一种新颖的虚拟化SMP机群的方法,即利用硬件虚拟化技术,在操作系统(OS, Operating System)之下构建分布式虚拟机监视器(DVMM, Distributed Virtual Machine Monitor),通过DVMM对机群的资源进行感知、整合、虚拟化和呈现,使SMP机群对OS呈现为一台具有共享内存特征的虚拟机;通过DVMM与OS配合,实现在虚拟机上透明地运行并行应用软件.

关 键 词: 虚拟化; 虚拟机监视器; 分布式虚拟机监视器; 对称多处理机; 机群

中图分类号: TP 316, TP 316.4

文献标识码: A **文章编号**: 1001-5965(2009)03-0300-04

Solution for SMP cluster virtualization

Peng Jinbing Long Xiang Xiao Limin

(School of Computer Science and Technology, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract Cluster is a kind of important parallel computer architecture with the advantages of low cost and good scalability, but also with the pitfalls of poor programmability and difficult administration. By means of the virtualization technologies, a distributed cluster could be abstracted to a share-memory virtual machine for supporting the share-memory programming mode, so that the pitfalls of a cluster could be overcome. Currently, there were only a few primary study results in this area, and they could not support the virtualization of SMP (symmetric multiprocessor) clusters. A novel solution was presented, which was constructing a DVMM (distributed virtual machine monitor) on a SMP cluster based on hardware-assisted virtualization technologies. The DVMM contained some symmetrical and cooperative VMMs distributed on multi-node. The cooperation among the VMMs virtualized the distributed hardware resources of the cluster. Thus, the DVMM could support an unmodified legacy OS (operating system) to run transparently on a SMP cluster. Compared with the related works, our solution has some advantages of good transparence, high performance, wide usage and easy implementation.

Key words virtualization; virtual machine monitor; distributed virtual machine monitor; symmetric multiprocessor; cluster

机群是一类重要的并行计算机体系结构,成本低廉,可扩展性强,但编程复杂,管理难度大[1].如何在保持良好可扩展性的同时,提高机群的可编程性和易管理性,是计算机界的研究热点之一.

虚拟化技术能够抽象计算机的物理资源,隔离硬件与软件[2].利用虚拟化技术可以将机群抽象成具有共享内存特征的虚拟机,支持共享内存

收稿日期: 2008-05-05

基金项目: 国家"863"高技术研究发展计划基金项目(2006AA01Z108)

作者简介: 彭近兵(1968-),男,安徽庐江人,博士生, pengjinbing@les.buaa.edu.cn

的编程模式, 从而克服机群的不足. 当前, 在该领域有少量探索性的研究成果, 但都不能支持对称多处理机 (SMP, Symmetric Multiprocessor) 机群虚拟化.

本文提出一种新颖的实现 SMP机群虚拟化的方法, 即利用硬件虚拟化技术, 在 SMP机群之上构建分布式虚拟机监视器 (DVMM, Distributed Virtual Machine Monitor), 通过 DVMM 实现 SMP机群虚拟化.

1　技术背景

1.1　虚拟化技术

虚拟化在计算机方面是指计算和处理在虚拟的基础上而不是真实的基础上运行. 虚拟化技术可在同一硬件平台上构建相互隔离的多个运行域, 每个运行域中可以运行自己的操作系统 (OS, Operating System) 和应用软件[2]. 按照实现方式不同, 虚拟化技术分为全虚拟化、泛虚拟化、预虚拟化和硬件虚拟化技术[3-4].

硬件虚拟化是最新的虚拟化技术. VT-x (Virtual Technology)[5] 是针对 IA-32的硬件虚拟化技术, 包括: 增加一种处理器工作模式, 称为虚拟机扩展 (VMX, Virtual Machine Extensions); 定义模式切换 VM entry (Virtual Machine entry) 和 VM exit, 增加控制结构 VMCS (Virtual Machine Control Structure); 增加 10条控制虚拟机的新指令. 在 VT-x支持下, 可以简化虚拟机监视器 (VMM, Virtual Machine Moniter) 设计, 实现全虚拟化.

1.2　相关的研究工作

目前, 关于机群虚拟化的研究刚刚起步, 相关的成果有 Virtual Multiprocessor[6] 和 vNUMA[7] 系统.

1.2.1　Virtual Multiprocessor

这是东京大学实现的 VMM. 它在 8台 PC机组成的机群上实现一台共享内存的 8路多处理器虚拟机. 该系统在主机 OS (Host OS) 支持下实现, VMM 运行在用户态; 采用泛虚拟化技术, 修改客户 OS; 用软件分布式共享存储 (DSM, Distributed Shared Memory) 实现共享内存; 通过进程模拟虚拟处理器; VMM 与 I/O服务器协作实现 I/O设备虚拟化. 该系统的不足是: VMM 实现在应用层, 性能低, 灵活性弱; 需要修改客户 OS代码, 实现难度大, 应用面窄; 只能利用一个结点的 I/O资源; 不能实现 SMP服务器机群虚拟化.

1.2.2　vNUMA

这是新南威尔士大学研制的 VMM. 它在 2台 IA64工作站组成的机群上实现一台 2-way非均匀存储访问 (NUMA, Nonuniform Memory Access) 虚拟机.

vNUMA 直接在硬件之上实现, 无须 OS支持; 采用预虚拟化技术, 在编译时静态地替换特权指令; 使客户 OS运行于 ring 1特权级, 以保证 VMM 的特权地位; 用 DSM 实现内存共享; 结点有主、从之分, 系统从主结点引导. 该系统的缺陷是: 需要修改客户 OS代码, 实现难度大, 应用受限; 降低客户 OS特权级, 易产生特权级混淆; 不能实现 SMP机群虚拟化.

2　SMP机群虚拟化方法

2.1　总体思路

本文的目标是将 SMP机群虚拟化为一台具有共享内存特征的虚拟机, 在其上透明地运行单一 OS和现有的并行软件, 从而克服机群难于编程和管理的缺陷. 实现此目标, 需要解决 4个问题: ①要能够感知机群的物理资源, 并进行整合以生成全局物理资源信息; ②要能够对全局物理资源进行虚拟化, 生成全局虚拟资源; ③要能够将全局虚拟资源信息呈现给 OS; ④要保证 OS能够像在单一 SMP机器上一样地管理和使用全局虚拟资源.

现有的技术不能实现这个目标. 1.1节介绍的各种虚拟化技术致力于在同一物理平台上构建多个相互隔离的虚拟机, 这与本文的目标相悖; 1.2节所述的 Virtual Multiprocessor 和 vNUMA 系统不能虚拟化 SMP机群.

本文提出一种实现上述目标的方法, 即通过 DVMM 实现 SMP机群虚拟化. 总体思路是: 基于硬件虚拟化技术, 在 SMP机群硬件之上构建一层系统软件, 称为 DVMM, DVMM 由分布于各结点之上的 VMM 组成, 各 VMM 完全对称; 在 DVMM 之上运行支持 cc-NUMA 的 OS; DVMM 感知整个机群的物理资源并分类整合生成全局物理资源信息, 虚拟化全局物理资源, 构建全局虚拟资源信息并呈现给 OS; OS基于所感知的虚拟资源集, 调度、执行进程, 管理、分配资源, 对底层的 DVMM 透明; DVMM 截获 OS访问资源的操作, 实现虚拟资源到物理资源的映射, 操纵物理资源, 并将执行结果反馈给 OS. 这样, DVMM 既能够将全局虚拟资源呈现给 OS, 又能够协助 OS管理和使用全局虚拟资源, 从而隐藏了底层硬件的分布式特性, 使机群对 OS呈现为一台 cc-NUMA 虚拟机, 即实现了 SMP机群的虚拟化.

2.2 关键技术

2.2.1 资源感知、整合与呈现

模拟并扩展 BIOS,成为扩展的基本输入输出系统(eBIOS, extended Basic Input/Output System). eBIOS在获得本结点资源信息之后,与其他结点的 eBIOS通信,收集机群的物理资源信息,并分类整合,生成全局物理资源信息.在保留各结点 VMM 自用的资源后,对余下的物理资源进行虚拟化,生成虚拟资源;组织虚拟资源,包括生成各类资源映射表,实现虚拟资源到物理资源的映射和物理资源到结点的映射;构建面向 OS的全局虚拟资源信息表.在虚拟资源的基础上加载 OS,截获 OS启动过程中的 BIOS调用,将全局虚拟资源信息反馈给 OS.

2.2.2 资源虚拟化

基于 eBIOS生成的全局物理资源信息实现资源虚拟化,包括指令集虚拟化、中断机制虚拟化、内存资源虚拟化和 I/O资源虚拟化.与单结点物理资源虚拟化方法[8-9]不同,本文采用的方法能够实现跨结点物理资源的虚拟化,如实现分布式内存资源的虚拟化.

利用 VT-x实现 IA-32指令集虚拟化,实现策略类似于 Xen的硬件虚拟机(HVM, Hardware Virtual Machine)[9].以软件模拟中断控制器:一方面,截获 OS对中断控制器的操作,若目标中断控制器在本结点,则读取或修改虚拟中断控制器的内容以反映客户操作的效果;若目标中断控制器在远程结点,则将操作请求发送到目标结点,目标结点的 VMM 操作虚拟中断控制器.另一方面,截获硬件中断,根据中断对象所在结点,分别由本结点的 VMM 或远程结点的 VMM 修改虚拟中断控制器的内容,将硬件中断呈现给 OS.结合影子页表[9]和软件 DSM 技术实现分布式内存虚拟化,即通过 DSM 将机群的内存资源整合成一个分布式共享内存,再通过影子页表技术虚拟化分布式共享内存.采用截获并代理执行 I/O操作的方法实现 I/O资源的虚拟化:通过 VT-x接管 OS的 I/O操作,若为本地 I/O操作,则 VMM 执行截获的 I/O指令,将结果反馈给 OS;若为跨结点的 I/O操作,则 DVMM 将 I/O指令传至宿主结点,宿主结点的 VMM 执行 I/O指令,将结果传回请求结点,请求结点的 VMM 将结果反馈给 OS.

2.2.3 资源管理与使用

OS负责管理和使用虚拟资源,DVMM 负责管理和使用物理资源,OS和 DVMM 通过 VM entry和 VM exit实现交互.OS基于虚拟资源集,独立地调度、执行进程,管理、分配虚拟资源,对底层的 DVMM 透明;当 OS执行敏感指令、出现中断或异常时,由 VM exit将系统控制权切换到 DVMM,DVMM 区分原因进行针对性地处理,如,将虚拟设备映射到物理设备,分配和操纵物理设备,通知中断信息等,然后通过 VM entry机制将结果反馈给 OS,同时将系统控制权交还 OS.通过 OS与 DVMM 之间的交互,实现对全局物理资源的管理与使用.

2.3 系统结构和运行机制

2.3.1 系统总体结构

如图 1所示,系统分 3层,自下而上分别为硬件层、DVMM 层和 OS层.硬件层包括若干个 SMP结点,结点的 CPU 支持 VT-x,各结点通过千兆以太网连接.DVMM 层由各结点上的 VMM 构成,各 VMM 通过专用底层通信软件进行通信.OS层可选用支持 cc-NUMA 的各类商用 OS.实现该系统的关键是构建 DVMM.下面分别介绍 DVMM 的结构和运行机制.

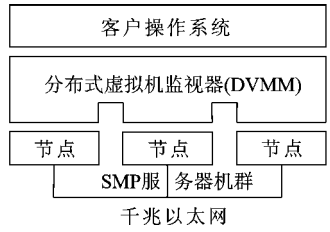

图 1 系统总体结构图

2.3.2 DVMM 结构

DVMM 运行于裸机之上,由分布于各结点的 VMM 组成,肩负资源感知、整合、虚拟化、呈现和跨结点协作等功能.DVMM 的功能结构如图 2所示.

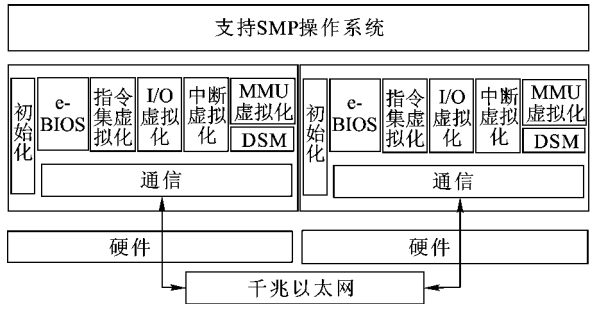

图 2 DVMM 结构图

初始化模块在各结点上引导 VMM. eBIOS模块感知、整合并向 OS呈现机群的资源配置.指令集虚拟化模块虚拟化 IA-32指令集,与中断虚拟化模块协作使 OS能够管理和调度机群的计算资源. I/O虚拟化模块虚拟全局 I/O资源,使 OS感知并使用机群的 I/O资源.中断虚拟化模块虚拟

化中断控制机制, 以虚拟方式向 OS通知中断事件. 存储器管理单元 (MMU, Memory Management Unit)虚拟化模块虚拟化内存资源, 确保 OS在虚拟化的物理地址空间中正确运行. DSM 模块将整个系统内存资源整合成一个分布式共享内存. 通信模块为各 VMM 提供低延迟、高带宽、可靠的通信服务.

2.3.3　DVMM 各模块间的关系及运行机制

DVMM 各模块之间的关系如图 3所示.

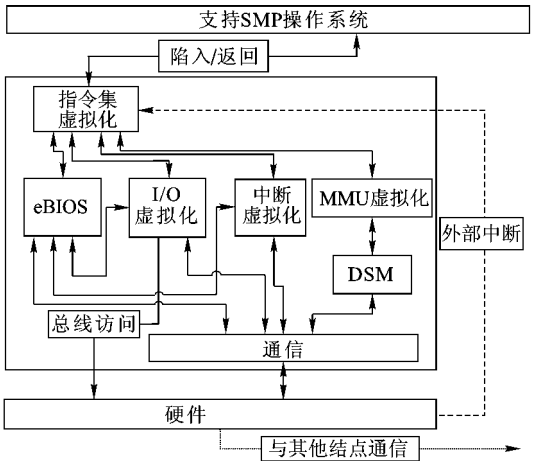

图 3　DVMM 各个模块关系示意图

指令集虚拟化模块是 DVMM 的入口点和出口点, 该模块可调用通信模块以外的所有模块, 也会被这些模块调用. 发生 VM exit时, 该模块分析陷入原因, 调用相应模块进行处理; 其他模块完成任务后, 调用该模块返回客户指令流. 通信模块是各 VMM 协作的基础, 该模块调用指令集虚拟化模块以外的所有模块, 也会被这些模块调用. eBIOS模块仅在初始化 DVMM 和启动 OS时起作用: 一方面 eBIOS模块通过中断虚拟化模块、I/O虚拟化模块、通信模块感知并生成整个系统的资源信息; 另一方面指令集虚拟化模块捕获 OS引导时的中断调用, 将信息传递给 eBIOS, eBIOS将系统虚拟资源信息反馈给 OS. I/O虚拟化模块接受指令集虚拟化模块的请求, 若为远程 I/O操作, 则调用通信模块, 向远程结点发出 I/O操作请求, 接收远程操作结果, 更新客户系统状态, 完成此次 I/O操作; 当 I/O 虚拟化模块接收到远程 I/O请求后, 对本地 I/O设备进行操作, 将结果传回请求结点. 指令集虚拟化模块调用中断虚拟化模块模拟 OS对虚拟中断控制器操作或者将外部中断向量转化为虚拟中断向量, 并向 OS注入一个虚拟中断. 当指令集虚拟化模块捕获 MMU 相关操作, 或者与 MMU 有关异常时, 调用 MMU 虚拟化模块进行处理. 当 MMU 虚拟化模块发现客户所请求的页面不在本结点时, 调用 DSM 模块进行页面迁移. 被 MMU 虚拟化模块调用时, DSM 模块向远程结点请求页面; 被通信模块调用时, DSM 模块对请求进行服务.

通过 DVMM 各模块协作, 实现了 SMP 机群资源的虚拟化, 使机群对 OS呈现为一台 cc-NUMA 虚拟机.

3　分析比较

由表 1可知, 与 VirtualMultiprocessor和 vNUMA 相比, DVMM 具有如下优势: ①DVMM 能够实现 SMP 机群虚拟化, 而 VirtualMultiprocessor和 vNUMA 都不能够实现; ②DVMM 利用最新的硬件虚拟化技术, 无需修改客户 OS, 设计和实现难度较低, 而 VirtualMultiprocessor和 vNUMA 分别采用泛虚拟化和预虚拟化技术, 都需要修改客户 OS, 实现难度高且应用受限; ③DVMM 在硬件辅助下实现, 运行在裸机之上, 因而性能高, 而 VirtualMultiprocessor和 vNUMA 都以软件实现, 性能较低, 并且 VirtualMultiprocessor实现在应用层, 穿越的软件层次多, 因而性能更低; ④DVMM 中的结点是对称的, 而 VirtualMultiprocessor和 vNUMA 中的结点都有主、从之分. 此外, DVMM 实现在系统软件层而 VirtualMultiprocessor实现在应用层, 因而 DVMM 在透明性方面优于 VirtualMultiprocessor; IA-32比 IA64应用广泛, 因而 DVMM 比 vNUMA 具有更广的应用前景.

表 1　DVMM 与 VirtualMultiprocessor和 vNUMA比较

系统	层次	技术	难度	透明性	对称	性能	SMP结点	架构
Multiprocessor	应用层	泛虚拟化	高	差	否	低	不支持	IA-32
vNUMA	系统软件层	预虚拟化	较高	好	否	较高	不支持	IA64
DVMM	系统软件层	硬件虚拟化	较低	好	是	高	支持	IA-32

4　结束语

本文基于硬件虚拟化技术在系统软件层将 SMP机群虚拟化为一台具有共享内存特征的虚拟机, 在其上透明地运行并行 OS和应用软件. 与同类技术相比, 本文所述方法具有支持 SMP结点, 实现难度适中、透明性好、性能高、各结点对称等优势, 更加适合于实现大规模服务器集群的虚拟化. 可以从以下方面开展进一步的研究: ①利用

(下转第 370页)

formicro inertialmeasurement uints[J]. Tsinghua Univ Sci& Tech, 2001, 41(8): 21-24(in Chinese)

[2] 杨华波, 蔡洪, 张士峰, 等. 高精度惯性平台连续自动标定自对准技术[J]. 宇航学报, 2006, 27(4): 600-604
YangHuabo, CaiHong, Zhang shifeng et al. Continuous calibration and alignment techniques for a high precision inertial platform[J]. Journal of Astronautics, 2006, 27(4): 600-604 (in Chinese)

[3] Mu Shuzhi, Bu Xiongzhu, Li Yongxin. Optimization design and calibration of installation error coefficients for gyroscope-free strapdown inertial measurement unit[C]// Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation. Luoyang: IEEE, 2006: 238-242

[4] Wang Jiandong, Liu Yunhui, Fan Weihong. Design and calibration for a smart inertial measurement unit for autonomous helicopters using MEMS sensors[C]// Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation. Luoyang: IEEE, 2006: 956-961

[5] Rogers R M. Applied mathematics in integrated navigation systems[M]. 2nd ed. Reston: AIAA, 2003: 173-183

[6] 杨晓霞, 黄一. 激光捷联惯导系统的一种系统级标定方法[J]. 中国惯性技术学报, 2008, 16(1): 1-7
Yang Xiaoxia, Huang Yi. Systematic calibration method for laser gyro SINS[J]. Journal of Chinese Inertial Technology, 2008, 16(1): 1-7(in Chinese)

[7] Grewal M S. Application of Kalman filtering to the calibration and alignment of inertial navigation systems[J]. IEEE, 1991, 36(1): 4-12

[8] Mirzaeri F M, Roumeliotis S I. A Kalman filter-based algorithm for IMU-camera calibration[C]//Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. San Digeo: IEEE, 2007: 2427-2434

[9] 陈哲. 捷联惯导系统原理[M]. 北京: 宇航出版社, 1986: 186-188
Chen Zhe. The principle of strapdown inertial system[M]. Beijing: SpaceNavigation Press, 1986: 186-188(in Chinese)

[10] Titterton D H, Weston J L. Strapdown inertial navigation technology[M]. England: Lavenham Press, 1997

(责任编辑: 吴 琼)

(上接第 303页)

设备虚拟化技术(VT-d[10], Virtualization Technndogies for directed I/O)和扩展页表(EPT, Extended page tables)[11]技术, 降低实现难度; ②以处理器一致性模型取代顺序一致性模型, 进一步提高性能; ③动态监测机群物理资源变化, 支持节点数随需增减; ④增加资源管理和作业调度功能, 支持多个运行域.

参考文献 (References)

[1] Culler D E, Singh J P, Gupta A. Parallel computer architecture-a hardware/software approach[M]. Beijing: China Machine Press, 1999: 20-37

[2] Smith J, Nair R. Virtual machines: versatile platforms for systems and processes[M]. Beijing: Publishing House of Electronics Industry, 2006: 3-13

[3] VMware: Understanding full virtualization, paravirtualization, and hardware assist[EB/OL]. 2007[2008-04-09]. http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf

[4] LeVasseur J, Uhlig V, Chapman M, et al. Pre-virtualization: slashing the cost of virtualization[EB/OL]. 2005[2008-04-09]. http://l4ka.org/publications/2005/previrtualization-tech-report.pdf

[5] Intel. Intel64 and IA-32 architectures software developer's manual[EB/OL]. 2007[2008-04-09]. http://www.intel.com/products/processor/manuals/

[6] Kaneda K, Oyama Y, Yonezawa A. A virtual machine monitor for providing a single system image[C]//In Proceedings of the 17th IPSJ Computer System Symposium (ComSys'05). Tokyo: Sur-Tokyo, 2005: 3-12

[7] Chapman N, Heiser G. Implementing transparent shared memory on clusters using virtual machines[C]//In USENIX Annual Technical Conference. Anaheim, 2005: 1-4

[8] Vmware. EXS server[EB/OL]. 2006[2008-04-09]. http://www.vmware.com/pdf/esx_datasheet.pdf

[9] Barham P, Dragovic B, Fraser K, et al. Xen and the art of virtualization[C]//In Proceedings of the 19th ACM SOSP, 2003: 164-177

[10] Intel. Intel Virtualization technology for directed I/O[EB/OL]. 2006[2008-04-09]. http://www.intel.com/technology/itj/2006/v10i3/2-io/7-conclusion.htm

[11] Gil Neiger. Intel virtualization technology: hardware support for efficient processor virtualization[J]. Intel Technology Journal, 3(10), 2006: 167-178

(责任编辑: 吴 琼)