面向计算系统的虚拟化技术

◆金 海 廖小飞

摘 要 本文介绍了虚拟化技术的发展历史,分类描述了虚拟化技术的现状,探讨了虚拟化所涉及的主要技术挑战:虚拟计算系统的体系结构、多计算系统虚拟化、用户使用环境的虚拟化、虚拟化系统的安全可信、虚拟计算系统性能评测的理论与方法等。最后介绍了973计划项目"计算系统虚拟化基础理论与方法研究"的最新研究进展。

一、引 言

早在20世纪50年代末,美国计算机学术界就开始了对虚拟技术的探索,首次提出了虚拟化的概念,但当时的虚拟技术只是应用在大型主机上。时至今日,伴随着IT硬件的丰富化、多样化以及一些软件公司如VMware、Xen、微软等推出不同的虚拟化软件之后,虚拟化的应用领域已经逐渐拓展,虚拟化技术也得到了大幅度提升。目前出现的许多不同种类的虚拟化解决方案,致力于从不同的角度解决不同的系统性能问题,使得虚拟化技术的内容越来越丰富。虚拟化生态链系统趋于标准化也体现了虚拟化技术的良好前景。

从最早的内存虚拟化到存储虚拟化,以及近年来大行其道的资源虚拟化和网格计算,虚拟化技术已经渗入到我们所触及IT领域中的各个层面。究其实质,虚拟化是为实现计算资源优化和体系结构透明化,而提供的人与计算机和谐的优化处理环境。正如汪成为院士所说,虚拟化成为对计算机"挖潜"和"优化"的首选途径。目前,虚拟化技术的研究主要侧重在系统级和网络级两个层面。以网格计算为代表的网络级虚拟化技术得到快速发展,它主要在网络计算中间件以及应用层进行研究与开发;而系统级虚拟化则侧重于计算机体系结构和底层系统软件,其目标是共享和整合广域分布的网络资源,为用户提供虚拟网络计算环境。近年来,两者有逐步融合的趋势。

本文作者:金 海、廖小飞,华中科技大学计算机学院,hjin@hust.edu.cn,xfliao@hust.edu.cn

研究资助:国家重点基础研究发展计划项目(2007CB310900)

目前所谈论的虚拟化技术,更多的是指计算系统虚拟化以及与之相关的虚拟机技术。近年来,学术界与工业界加大了对虚拟化技术的关注力度。国际著名科技咨询机构Gartner在2006和2007年连续两年把虚拟化技术预测为未来最关键的10大IT技术之首。诸如VMWare、XenSource等在内的一大批新兴公司脱颖而出,而其他老牌软硬件厂商也相继推出了自己的虚拟化产品。

关于虚拟化(Virtualization)的定义有多种不同的看法。G. Popek和R. Goldberg提出[1],虚拟化应具有以下3个特点:(1)保真性(Fidelity)——强调应用程序在虚拟机上执行,除了时间因素外(会比在物理硬件上执行慢一些),应与在物理硬件上具有相同的执行行为;(2)高性能(Performance)——强调在虚拟执行环境中应用程序的绝大多数指令能够在虚拟机管理器不干预的情况下,直接在物理硬件上执行;(3)安全性(Safety)——物理硬件应该由虚拟机管理器全权管理,被虚拟出来的执行环境中的程序(包括操作系统)不得直接访问硬件。

本文采用一种广义的虚拟化定义:虚拟化是一种采用软硬件分区、聚合、部分或完全模拟、分时复用等方法来管理计算资源、构造一个或多个计算环境的技术。当前,系统虚拟化技术面对着新的软硬件环境,存在诸多问题,同时也存在着各种挑战和机遇。本文从多个角度阐述虚拟化技术的现状、面临的挑战以及973计划项目"计算系统虚拟化基础理论与方法研究"的最新研究进展。

二、虚拟化技术的发展历史及现状

虚拟化技术的起源可以追溯到20世纪50年代末,1959年计算机科学家Christopher Strachey发表了一篇名为"大型高速计算机中的时间共享"(Time Sharing in Large Fast Computers)的学术报告[2],他在文中首次提出了虚拟化的基本概念,被认为是虚拟化技术的最早论述。在虚拟化技术萌芽初期,人们研究虚拟计算的目标是对相对昂贵的硬件资源进行

充分利用,通过虚拟机手段让更多的人能够通过终端设备接触和使用计算机系统。虚拟机技术的发明能够让更多的用户更好地共享当时非常昂贵的计算机资源。但是到20世纪70—80年代,随着大规模集成电路的出现和个人电脑的普及,计算机硬件变得越来越便宜。当初为共享昂贵硬件而设计的虚拟化技术慢慢无人问津,而只是在高档服务器(如IBM小型机)中继续存在。

近年来,虚拟机以及计算虚拟化技术重新成为计算机体系结构领域的研究热点,其原因主要有以下两个方面:其一,计算机系统经过多年发展,在变得越来越强大的同时,也在变得越来越难以管理(例如今天的网络系统、分布式计算系统),软硬件(TCO)管理开销(特别是电费开销)也逐年增加。特别是随着处理器多核化时代的到来、冗余计算资源的引入,这一矛盾势必越来越尖锐。其二,今天的计算已经从以前的以计算机为中心向以用户为中心的服务计算过渡,人们更关心的是计算系统能够为用户提供怎样的接口和提供怎样的服务,以适应用户复杂和多样化的需求。在这一背景下,由于计算系统虚拟化技术既能够屏蔽底层复杂的物理环境,又能够为用户提供可配置的使用环境,就自然重新成为工业界和学术界的研究热点。虚拟化技术在层次上可划分为:指令集架构级、硬件级、操作系统级、编程语言级和数据库级虚拟化。

1. 指令集架构级虚拟化

指令集架构级虚拟化通过纯软件方法,模拟出与实际运行的应用程序(或操作系统)所不同的指令集去执行,采用这种方法构造的虚拟机一般称为模拟器(emulator)。一个典型的计算机系统由处理器、内存、总线、硬盘驱动器、磁盘控制器、定时器、多种I/O设备等部件组成。模拟器通过将客户虚拟机发出的所有指令翻译成本地指令集,然后在真实的硬件上执行。这些指令包括典型的处理器指令和特殊的I/O指令。这种类型的虚拟机结构,除具有简单性和鲁棒性特点外,它能够实现应用以及操作系统的跨平台执行。由于模拟器的工作原理是将虚拟执行环境中的指令翻译成主机平台上的指令来执行,当客户平台的架构改变时,它也可以很容易地适应平台的变化。这样,它在客户平台和主机平台之间就没有执行严格的绑定。例如,模拟器能够让原来运行在x86结构上的操作系统和应用程序在PowerPC系列处理器、甚至是在为嵌入式系统而设计的Arm系列处理器上运行。然而,架构的便携性也带来一些性能上的开销。因为由仿真计算机发出的每一条指令都需要用软件方法来解释,这就导致较大的性能损失。目前比较典型的模拟器系统有Bochs[3]、VLW[4]、QEMU[5]和BIRD[6]等。

2. 硬件级虚拟化

硬件抽象层面(hardware abstraction layer, HAL)虚拟化实际上与指令集架构级虚拟化非常相似,其不同之处在于,这种类型的虚拟化所考虑的是一种特殊情况:客户执行环境和主机具有相同指令集合,并充分利用这一特点,让绝大多数客户指令在主机上直接执行,从而大大提高了执行的速度。目前,大部分商业虚拟化软件在流行的x86平台使用此虚拟化技术来提高效率,表明了这类虚拟化技术的可行性与实用性。该虚拟化技术可以将虚拟资源映射到物理资源并在虚拟机计算中使用本地硬件。当虚拟机需要访问关键物理资源时,模拟器接管其物理资源并妥善地多路复用。这种虚拟化技术需要所构造的虚拟机能对其中的一些特权指令(例如修改页表等操作)进行处理,执行时产生陷入并将它传递给下层虚拟机管理器(virtual machine monitor, VMM)执行。这是因为在虚拟机中运行的未加修改的操作系统会利用特权指令得到CPU和内存资源。当某特权指令执行时产生一个陷入,便马上将指令发送给VMM,这使得VMM可以完全控制虚拟机并保持每个虚拟机隔离。然后,VMM在处理器中执行该指令,并将模拟结果及特权指令返回给虚拟机。大多数商业虚拟机软件,都使用像代码扫描和动态指令重写这样的技术来解决这些问题。硬件层次的虚拟机具有高度的隔离性(虚拟机和底层物理机器均实现隔离)、易于被用户接受(和用户所习惯使用的普通机器看起来一样)、支持不同操作系统和应用程序、低风险和易于维护等特点。目前,较为典型的系统包括VMware、Virtual PC、Denali[7]、Xen[8]、KVM、UML inux[9]和CoL inux[10]等。

3. 操作系统级虚拟化

由于硬件层次的虚拟机可以直接对裸机进行访问,从而导致用户在调试或运行应用程序之前需要花费大量时间来安装和管理虚拟计算机,其中包括操作系统安装、应用套件安装以及网络配置等等。如果用户希望虚拟机中所具有的操作系统和物理机器上安装的一样,并利用该虚拟环境来进行一些安全或沙盒测试等方面的实验,就需要操作系统级虚

拟化技术。操作系统级虚拟化技术的关键思想在于,操作系统之上的虚拟层按照每个虚拟机的要求为其生成一个运行在物理机器之上的操作系统副本,从而为每个虚拟机产生一个完好的操作环境,并且实现虚拟机及其物理机器的隔离。通过操作系统虚拟化方式,可以有效避免对物理机器的重复安装,从而减少用户在机器安装上所花费的时间和精力。该层次的虚拟机和底层物理机器上的操作系统共享硬件资源,并在操作系统之上通过一个虚拟层(类似于 VMware的虚拟机管理器)来展现给用户多个独立的、隔离的机器。一个应用的操作环境包括操作系统、用户函数库、文件系统、环境设置等。如果应用系统所处的这些环境能够保持不变,那么应用程序自身无法分辨出其所在的环境与真实环境之间的差别。目前,典型系统有 Jail[11]、Asplinux和 Ensim的虚拟个人服务器(VPS)等。

4. 编程语言级虚拟化

传统机器通过 ISA(instruction set architecture)的支持来执行指令集。正是由于 ISA抽象层的存在,使得在机器上运行的操作系统和程序相当于机器上的应用程序。硬件的操作通过专用的 I/O指令(I/O映射),或者将内存分配给 I/O然后再操作内存(内存映射)的方式来进行处理。但无论如何,应用程序最终还是由一系列的指令所组成。随着 Java虚拟机(JVM)的到来,使得这种新的实现虚拟机的方式逐渐引起人们的注意。这种抽象层次的虚拟化技术的主要思想是在应用层次上创建一个和其它类型虚拟机行为方式类似的虚拟机,并支持一种新的自定义指令集(例如 JVM中的 Java字节码)。这种类型的虚拟机使得用户在运行应用程序的时候就像在真实的物理机器上一样,并且不会对系统的安全造成威胁。像普通的机器一样,它通过安装一个商业的操作系统或利用其自身的环境来为应用程序提供操作环境。这种抽象层次的虚拟化系统主要有 Java虚拟机、Microsoft. NET CLI和 Parrot等。

5. 程序库级虚拟化

在几乎所有的系统中,应用程序的编写都使用由一组用户级库来调用的 API函数集。这些用户级库的设计能够隐藏操作系统的相关底层细节,从而降低普通程序员的软件开发难度。该层次虚拟化技术工作在操作系统层上,创造了一个与众不同的虚拟环境,在底层系统上实现了不同的应用程序二进制接口(ABI)和不同的应用程序编程接口(API)。这种技术能很好的完成 ABI/API仿真工作。该类系统的典型代表有 Wine等。

三、虚拟化技术的相关研究内容

面向计算系统的虚拟化技术则呈现了一些新的趋势,构造涵盖用户级、服务器级虚拟化为一体,可支撑单机环境、多机环境的安全轻量级的虚拟环境成为一个重要挑战,它涉及虚拟计算系统的体系结构、多计算系统的虚拟化、用户使用环境的虚拟化、虚拟化系统的安全可信、虚拟计算系统的性能评测的理论与方法等方面。

1. 虚拟计算体系结构

系统级虚拟化在计算机硬件和操作系统之间增加虚拟机管理器(VMM)以解除二者间的直接依赖。按照 Popek和 Goldberg制定的虚拟化准则,大多数现代 CPU体系结构是没有设计成可虚拟化的,包括最流行的 x86体系结构。提供虚拟化能力的最直接方法是修改操作系统,将原来指令集中不能虚拟化的部分替换成容易虚拟化的和更高效的等价物,这种方法通常称为部分虚拟化(para-virtualization),英国剑桥大学开发的 Xen虚拟机就是采用这种方法[8]。为了提供快速、兼容的 x86体系结构的虚拟化,VMWare采用了全虚拟化(full-virtualization)的技术路线,它将传统的直接执行和快速的动态二进制翻译技术结合起来,利用二进制翻译器运行不能虚拟化的特权模式,补偿不能虚拟化的 x86指令。德国 Karlsruhe大学、澳大利亚新南威尔士大学和 IBM的研究人员共同提出了预虚拟化(pre-virtualization)方案,将操作系统中的特权指令静态替换为虚拟层的接口调用,这样无需修改源代码即可使客户操作系统支持虚拟化。

随着硬件技术的发展,硬件本身也为虚拟化提供了支持。国际主流微处理器厂商也积极开展虚拟化相关研究,并推出支持硬件辅助虚拟化的产品和系统。Intel推出了 VT虚拟化技术,它包括支持指令集虚拟化的 VT-x和 VT-i技术以及支持 I/O设备虚拟化的 VT-d技术。硬件辅助虚拟化的支持简化了 VMM的设计和实现,有利于提高虚拟机的性能。AMD公司也推出类似的硬件辅助虚拟技术 Pacifica。目前硬件辅助虚拟化技术还需进一步完善,需要在理论、模型和实验等层面进行深入研究。

半虚拟化、全虚拟化、预虚拟化、硬件支持虚拟化等各有优势和不足,如何融合各种虚拟化方法的

优势,按照应用任务的需求,将资源进行共享和动态划分,使计算系统具备动态构建能力,这是需要深入研究的问题。

2. 多计算系统的虚拟化

从现有的多机环境虚拟化研究状况来看,计算系统体系结构的紧耦合特性与多粒度资源使用需求之间的矛盾凸显出了多计算系统的动态构建问题。如何从系统结构的角度,按照应用任务的需求,将资源进行共享和动态划分,以便于动态建立基于多核的虚拟计算机或者基于分布式计算资源的虚拟计算环境,显得尤为重要。而由于系统可被多个用户同时使用,如何协调多个用户的请求,优化系统性能、降低系统运行成本也就成为动态构建技术的主要研究内容。值得指出的是,为了达到先进的计算环境特性,支持更加广泛的应用,在动态构建问题上虚拟机的迁移问题成为关注的焦点。

虚拟机的迁移分为动态迁移和静态迁移,其中动态迁移能够在将虚拟机中运行的操作系统与应用程序从一个物理节点迁移到另外一个运行节点的过程中,保持客户操作系统和应用程序的继续运行而不受干扰。因此,更能够实现服务器之间保持负载均衡、保证服务质量、节约电能等诸多优秀的特性。目前,很多机构都在研究新型系统级虚拟化架构和虚拟计算系统动态构建的理论及优化方法,但是这些方法多数并不适用于跨物理主机的集群系统。此外,有不少机构研究了虚拟机在分布式环境下的应用。剑桥大学基于 Xen的 Parallax项目可以管理大量虚拟机,通过消除写共享、增加客户端缓存、利用模版映像来构建整个系统。Ventana系统利用集中存储来保证虚拟服务的多版本、隔离性和移动性。美国 Florida大学的研究人员首次提出将传统虚拟机应用于网格等分布式环境,并提出了基于虚拟机的网格服务体系结构。目前,国际上基于虚拟机的分布式计算已经成为了研究的热点[12, 13]。相对于单一计算节点来说,由多个节点构成的多计算系统的管理是一个挑战。因为其管理难度不再是对虚拟机和实际机器的识别和管理,而是如何在更大规模的系统上实现更多的虚拟机,以更出色的整合能力使用户通过简单的操作界面统一管理几个数据中心的资源。为了解决这个问题,引入了多虚拟机的单一映像方法,使得多个计算系统资源看起来就像一个单一的计算对象,从而将多计算系统转化成单一计算系统。但是此类系统的扩展性和性能总体不佳。

3. 虚拟用户使用环境

虚拟化技术的根本目的是为了提高普通用户对计算机的操控力,降低计算机使用的复杂性,最终提高利用效率。虚拟用户使用环境是指将虚拟化的理念、技术广泛应用于桌面环境、用户操作环境,建立一套可移植、可重构、按需定制的可视化用户使用环境和程序执行的自动配置环境,以适应软硬件环境和任务需求的变化,协调分布呈现的计算资源,最终建立任务执行的协同计算环境。由此可以看出,建立虚拟化的用户使用环境是一项综合性课题,涉及到桌面虚拟化技术、应用程序的虚拟化、网络虚拟化和虚拟化编程环境等。

桌面虚拟化改变了传统的计算机管理方式,利用虚拟化技术对操作系统及应用程序进行集中的管理和高效的分发迁移,使得用户在任何时间、任何地点,只要具备基本的硬件就可以使用自己需要的工作环境。应用程序虚拟化则可以把应用程序从操作系统中解放出来,实现自给自足的虚拟运行环境。在成功实现这种分离后,还产生了很多优化技术,如将应用软件流水化包装起来,应用软件无需完全安装,只要一部分程序能够在电脑上运行即可。当前,该领域研究人员已经开展了一些卓有成效的研究工作,包括 YouOS、eyeOS、卡耐基梅隆大学和 Intel合作的 ISR (Internet Suspend/Resume)、斯坦福大学的 Collective项目[14]等。但是,系统的透明性、用户接口的平台无关性、用户需求的随处可满足性仍无法充分做到。用户还不得不与各类不同的用户接口、系统直接打交道,工作效率不能有效地提高。如何构造一个真正的可移植、可重构、按需配置的普适运行环境尚在进一步的研究中。

4. 虚拟化系统的安全可信机制

归结起来,虚拟化系统的安全挑战主要有以下两个方面:一是计算系统体系结构的改变。虚拟化技术已从完全的物理隔离方式发展至共享式虚拟化,实现计算系统虚拟化需要在计算性能、系统安全、实现效率等因素间进行合理的权衡。在这种过渡条件下,虚拟机监视器和相关具有部分控制功能的虚拟机成为漏洞攻击的首选对象,使之成为最重要的安全瓶颈。二是计算机系统的运行形态已经发生改变。虚拟计算允许用户通过操纵文件的方式来创建、复制、存储、读写、共享、移植以及回滚一个机器的运行状态,这些虽然极大地增强了使用的灵活性,但却破坏了原有基于线性时间变化系统设定的

安全策略、安全协议等的安全性和有效性,这包括软件生命周期和数据生命周期所引起的系统安全[15]。在虚拟机的安全验证方面,最典型的代表是美国密西根大学的 Peter Chen和 Brian Noble研发的 ReVirt系统,该系统采用了反向观察点和反向断点技术针对虚拟机上的恶意攻击进行检测和回放,提供验证功能[16]。

除了上述研究之外,基于虚拟机的入侵检测技术也作为与虚拟机技术相辅相成的手段发展起来。在运行虚拟机的宿主主机平台上建立入侵检测系统,为网络系统的安全提供了保障。在入侵检测技术中,传统的蜜罐技术和蜜网技术也都可以通过虚拟化技术实现,是目前基于虚拟机的入侵检测系统的主要应用。综合而言,当前开展的虚拟机安全性研究大多都针对某一个安全问题进行,系统性的分析与研究仍在不断探索。

5. 虚拟计算系统的性能评测

从上文所述的相关技术内容来看,虚拟计算系统已越来越多地为一些重要的商业应用提供了行之有效的解决方案。这些解决方案带来了更多的复杂性和性能开支,使得针对虚拟计算系统的性能评测变得尤为重要。

虚拟计算系统的评测内容包括系统性能和可用性评测等。目前,可用于虚拟计算系统性能评测的方法主要是借鉴传统计算系统已有的性能评测方法——测量方法、模拟方法和分析方法。测量方法通过运行涉及不同类型计算的基准测试程序(benchmark)评测系统的计算能力。分析方法通过为计算系统建立数学模型,进而在给定输入条件下通过计算获得目标系统的性能特性[17, 18]。然而,计算资源的虚拟化和计算资源聚合的动态化使传统计算系统的评测研究方法并不能完全适应虚拟系统的评测,需要面向计算资源虚拟化和动态构建特征逐步确立适合自身的虚拟计算系统评测理论与方法。因此,在进一步研究中,需要针对不同计算性质的任务模式、应用测量方法和统计分析技术,研究典型虚拟机不同技术途径在性能上的表现特征及性能瓶颈,分析提高性能的调优策略。

四、计算系统虚拟化基础理论与方法研究进展

"计算系统虚拟化基础理论与方法研究"项目是973计划在2007年部署的关于计算系统虚拟化的研究项目。研究团队由华中科技大学牵头,由北京大学、清华大学、上海交通大学、国防科技大学、浙江大学、江南计算技术研究所和航天科工集团第二研究院的研究人员组成。该项目希望从资源使用环境、任务执行环境以及用户操作环境出发,探讨在计算系统各个层面实现虚拟化技术的理论、机制及方法。该项目包括8个课题,分别为:(1)计算系统虚拟化理论模型及体系结构;(2)单计算系统资源虚拟化方法;(3)多计算系统资源虚拟化方法;(4)虚拟计算系统普适化运行环境;(5)虚拟计算系统安全可信机制;(6)虚拟计算系统评测理论与方法;(7)基于高效能计算机的虚拟化技术;(8)虚拟化仿真系统应用。

项目研究工作主要集中在:虚拟化体系结构设计;指令集的模拟和二进制翻译;计算环境的快速部署;多机虚拟化的资源监控和统一管理;用户使用环境的迁移、重构、协同;程序运行环境的动态按需配置;虚拟机安全隔离;虚拟计算系统性能评测;面向高效能计算机和大型仿真应用的验证研究等。

目前,项目研究团队基于已有的研究基础,已经在二进制翻译、外存虚拟化、虚拟机迁移、多机内存虚拟化、多虚拟机管理、虚拟工作环境等多个方面取得了重要进展,为后续研究工作打下了坚实的基础。已经在包括 Cluster2008、DAC2008、SVM2008、《软件学报》、《电子学报》等在内的一些重要国际国内期刊及本领域重要国际学术会议上发表了近30篇学术论文。本文主要选取一些系统原型的研究进展加以描述,而在具体算法、机制等方面的进展信息可参考该项目的开放信息平台(http://grid.hust.edu.cn/973)。

1. 虚拟机基础研究

(1)内存虚拟化

项目提出了 VMM 进行内存管理的一种机制——虚拟机的动态内存映射,它允许 VMM 在虚拟机运行时,动态地改变它的物理内存与机器物理内存的映射关系。利用该机制,VMM 能够方便地实现按需取页、页面交换、Ballooning、内存共享、Copy-On-Write等虚拟机高级内存管理技术,具有很强的适用性和可扩展性。并在一个开源的虚拟机管理器 KVM(kernel based virtual machine)上实现了动态内存映射机制。测试表明,该机制能够在充分保证虚拟机访问内存性能的前提下,实现虚拟机内存的动态管理和调配。此外,还利用该机制在 KVM 上实现了虚拟机的远程内存交换。

(2) CPU虚拟化

项目在 Intel VT硬件平台上设计了一个轻量级的 VMM,以实现 CPU 的虚拟化。轻量级是指该 VMM结构简洁,充分利用硬件的虚拟化扩展,实现 CPU的虚拟化。该 VMM采用简单的内存管理模型,预先为每个虚拟机分配一段连续物理内存,不考虑虚拟机的 I/O。该轻量级 VMM主要目的是研究在具有虚拟化扩展的硬件平台上如何实现 CPU的虚拟化,并充分利用现有硬件虚拟化支持,对虚拟机运行环境进行细粒度的配置,实现性能优化,包括设置 Exception Bitmap、I/O Bitmap、CR寄存器的访问权限位等,权衡隔离性和性能,尽量降低 VM Exit的频率。通过实现和测试,分析现有硬件虚拟化支持的优势和不足,例如现有硬件平台上虚拟化的代价,包括环境切换频率和状态保存/恢复的开销。此外针对多核的发展趋势,比较分时复用和分派核心两种虚拟化方式的性能。

(3) 基于虚拟机回放的在线迁移

目前的 Live Migration (VMotion &XenMotion)大多采用 Memory-to-Memory的方法,只能运用于集群系统,应用场景有限,而且还存在以下问题:迁移时若内存修改频率过快,超过了内存页面传输的速度,停机时间 (downtime)还比较长 (内存修改频率高时可达 3.5s);迁移过程总体传输的数据量很大,需要消耗的网络带宽也很大;利用 NAS或 SAN的共享来解决磁盘迁移,没有解决使用 Local Disk的磁盘迁移;只适用于 LAN (网络连接保持在 MAC层)。而项目研究基于源主机上 Log数据产生速度要比内存页面修改速度慢得多,形成了利用 checkpoint/recovery结合 trace/replay技术来实施虚拟机在线迁移的思路。由于最后一轮的 Log传输数据非常小,停机时间可以减小到毫秒级。

2. 虚拟机快速部署系统

VNIX管理系统是多计算系统资源虚拟化技术上的一个典型应用,其目标是创建一个更好的多机系统管理模式,服务于多计算系统虚拟化领域内的科学研究。项目通过对 VNIX管理系统应用需求进行分析,在集群环境和虚拟机软件 Xen平台上,开发出一组适合用户公共需求的服务以做到对海量数据存储、高性能计算、分布式资源管理和信息服务、资源和设备的整合以及对资源的高效透明使用。并基于这些服务,针对不同的应用,构建出适应该应用的虚拟科研环境;可根据各用户对问题解决环境的使用需求的不同,提供灵活的用户管理机制及安全、方便的使用接口。

为了提供灵活、可扩展的基于虚拟机的管理体系,VNIX管理系统采用 4层体系结构:(1) 目标系统层,即集群节点机上的各种计算资源以及 VNIX管理系统所依赖的虚拟机软件平台 Xen或 QEMU;(2) 基础服务层,包括单一计算节点上的虚拟机管理、虚拟机信息提取、虚拟机创建等核心模块,它们是上层核心服务的基础;(3) 核心服务层,该层统一管理全部计算节点,包括虚拟机快速部署、虚拟机状态监控、资源动态调整、负载均衡实现 4个核心模块,这 4个核心模块可移植性和可扩展性都较强,可以移植到其它的集群系统中;(4) 用户接口层,该层通过客户端界面或者 Web界面把底层服务运行的结果返回给用户 (用户包括管理员、注册用户以及游客)。

3. 虚拟工作环境

VirtualDesk系统是一个集成异构平台下的应用程序资源为用户构建工作平台的工具。VirtualDesk系统目前支持 Window和 Linux系统。用户使用 VirtualDesk客户端连接到服务器,就可以使用由服务器端管理的其它多个服务器上的应用程序。用户通过点击服务器提供的应用程序列表中的项,就可以打开应用程序,在本地看到应用程序的窗口,并像操作本地程序一样操作服务器上的程序。使用 VirtualDesk上的应用程序可以不用安装服务器上已有的软件,大大减轻了构建用户工作环境的负担。VirtualDesk支持一个客户端同时连接多个服务器,打开多个应用程序界面。

目前已经实现异构平台程序资源的窗口级混合,屏蔽了操作系统的异构性。下一步,该虚拟工作环境将实现运行时迁移、复制、消亡及快速启动等一体化功能,并重点研究基于虚拟机的动态资源分配、虚拟环境的安全维护机制、多虚拟工作环境的协作与分发等重要问题。

4. 虚拟集群管理与维护

现有动态集群的研究主要试图解决集群基础架构和应用程序需求之间的矛盾,包括:(1) 共享的物理集群环境和用户不同的软件环境需求之间的矛盾,采用按需配置的动态集群 (cluster on demand, COD)能够在一定程度上解决这个矛盾,但物理集群部署和重新部署的开销较大,而且造成了物理节点之间的隔离,难以充分利用物理资源;(2) 多任务共享物理集群环境的调度问题,例如在同一个物理集

群中,并行应用和串行应用的调度矛盾;(3)由于集群之间可能的异构性,难以利用多个集群共同处理较大规模的任务。基于现有研究,项目采用虚拟机的 Suspend/Resume及 Ballooning等技术,提出了一套虚拟集群按需创建、动态部署、高效分配的管理工具,从而提高了虚拟集群环境重新部署性能,减小了物理节点在虚拟集群之间的转移代价。

项目初步完成了多机虚拟化环境的构造工具,可以用于快速构建面向高性能集群的多机虚拟化环境。目前,该工具已具有的能力包括:(1)构建虚拟集群——已经实现了一个可跨平台运行的控制台,可以实现快速虚拟机、虚拟集群与 VMM配制的建立、删除、导入和导出,并控制它们的运行机制与状态,还实现了集群管理的批量操作,建立与配制可以快速完成;(2)对虚拟机集群中节点的远程控制——通过将 VNC集成到控制台,可以远程集中控制虚拟计算节点;(3)异构环境的支持——控制台可以运行在 Linux与 Windows平台,虚拟高性能计算环境可以运行在多种 VMM上,同时虚拟高性能计算环境也可以包含多种平台(Linux、Windows等),高性能计算环境通过自定义的 XML来描述,并在控制层提供对异构的支持;(4)虚拟高性能计算环境监测——将物理节点与虚拟节点采样数据汇总到控制台,并以图形的方式给出结果。

五、结 语

虚拟化技术可将底层物理设备与上层操作系统、软件分离并去耦合,从而实现 IT资源利用效率和灵活性的最大化。汪成为院士曾在《计算机学会通讯》上撰文指出,虚拟化技术是需求牵引、技术推动的产物,是对计算机系统"挖潜"和"优化"的首选途径。他还指出,强调虚拟化是为了提高实效,虚拟化是为了对所研究的问题描述得更本质化,在处理该问题时更简捷化。秉承此念,还需继续努力,在解决计算机体系结构领域的这一重要挑战上走得更远。

参考文献

[1] Popek G, Goldberg R. Formal requirements for virtualizable third generation architectures. Communications of the ACM, 1974, 17(7): 413—421

[2] Strachey C. Time sharing in large fast computers. Proceedings of the International Conference on Information Processing, UNESCO, June 1959

[3] Lawton K, Denney B, Guarneri N D, Ruppert V, Bothamy C, Calabrese M. Bochs x86 pc emulator users manual. http://bochs.sourceforge.net/, 2003

[4] Transmeta. Crusoe processor. http://wwww.transmeta.com, 2000

[5] Bellard F. QEMU: A fast and portable dynamic translator. Proceedings of the USENIX Annual Technical Conference, 2005: 41—46

[6] Nanda S, Li W, Lam L C. Binary interpretation using runtime disassembly. http://www.ecsl.cs.sunysb.edu/bird/index.html, 2005

[7] Warfield A, Ross R, Fraser K, Limpach C, Hand S. Parallax: Managing storage for a million machines. Proceedings of USENIX Hot Topics in Operating Systems (HOTOS), June, 2005

[8] Barham P, Dragovic B, Fraser K, Hand S, Harris T, Ho A, Neugebauer R, Pratt I, Warfield A. Xen and the art of virtualization. Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP), 2003: 164—177

[9] Dike J. A user-mode port of the linux kernel. Proceedings of the 4th Annual Linux Showcase and Conference, 2000

[10] Alon D. Cooperative linux. Proceedings of the Linux Symposium, July 2004: 23—31

[11] Kam PP, Watson R. Jails: Confining the omnipotent root. Proceedings of the 2nd International SANE Conference, 2000

[12] Whitaker(A) Cox R S, Shaw M, Gribble S (D) Constructing services with interposable virtual hardware. Proceedings of the First Symposium on Networked Systems Design and Implementation (NSDI'04), San Francisco, California, March 2004.

[13] Massie M L, Chun B N, Culler D E. The ganglia distributed monitoring system: design, implementation, and experience. Parallel Computing, 2004, 30(7): 817—840

[14] Govil K, Teodosiu D, Huang Y, Rosenblum M. Cellular Disco: Resource management using virtual clusters on shared-memory multiprocessors. Proceedings of 17th Symposium on Operating Systems Principles, 1999

[15] Chandra R, Zeldovich N, Sapuntzakis C, Lam M S. The Collective: A cache-based system management architecture. Proceedings of the Second Symposium on Networked Systems Design and Implementation (NSDI2005), May 2005: 259—272

[16] Dunlap G W, King S T, Cinar S, Basrai M, Chen P M. ReVirt: enabling intrusion analysis through virtual-machine logging and replay. Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI'02), ACM Press, Boston, MA, USA, December 8-11, 2002: 211—224

[17] Adve V S, Vernon M K. Parallel program performance prediction using deterministic task graph analysis. ACM Trans. Comput. Syst., 2004, 22(1): 94—136

[18] Chen S, Liu Y, Gorton I, Liu A. Performance prediction of component-based applications. Journal of Systems and Software, 2005, 74(1): 35—43

Virtualization Technology for Computing System

Jin Hai, Liao Xiaofei

School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074