ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.8, August 2007, pp.2016−2026
DOI: 10.1360/jos182016
© 2007 by *Journal of Software*. All rights reserved.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于虚拟机的虚拟计算环境研究与设计*

怀进鹏, 李 沁+, 胡春明

(北京航空航天大学 计算机学院,北京 100083)

Research and Design on Hypervisor Based Virtual Computing Environment

HUAI Jin-Peng, LI Qin+, HU Chun-Ming

(School of Computer Science and Engineering, BeiHang University, Beijing 100083, China)

+ Corresponding author: Phn: +86-10-82339284, Fax: +86-10-82316796, E-mail: liqin@act.buaa.edu.cn, http://act.buaa.edu.cn

Huai JP, Li Q, Hu CM. Research and design on hypervisor based virtual computing environment. *Journal of Software*, 2007,18(8):2016−2026. http://www.jos.org.cn/1000-9825/18/2016.htm

Abstract: Based on the analysis of hypervisor technologies and typical projects on the hypervisor based virtual computing environment, the design of CIVIC (CROWN-based infrastructure for virtual computing) is provided, which has three characteristics. Firstly, it can offer separated and isolated computing environment for users. Secondly, it can also realize hardware and software consolidation and centralized management for computer administrators. Thirdly, it can be transparent to upper layer applications, hiding the dynamicity, distribution and heterogeneousness of underlying resources from applications. The experimental results show that CIVIC can provide a customized computing environment for user easily and efficiently.
Key words: virtual computing; hypervisor; virtualization; grid computing

摘 要: 通过对基于虚拟机的虚拟计算环境典型系统的分析,给出了 CROWN 虚拟计算平台 CIVIC(CROWN-based infrastructure for virtual computing)的设计.CIVIC 集成多种虚拟机技术,可以为用户提供独立、隔离的计算环境,可以为管理人员提供硬件资源和软件资源的集中管理功能,支持对应用程序的透明性,屏蔽底层硬件资源的动态性、分布性和异构性.实验结果表明,CIVIC 能够方便、高效地为用户定制所需计算环境.
关键词: 虚拟计算;虚拟机;虚拟化;网格计算
中图法分类号: TP393 **文献标识码**: A

随着计算机和互联网技术的不断发展及应用的深入,网络已聚合了各种计算资源、数据资源、软件资源以及服务资源等,但存在总量丰富但资源利用率低的矛盾.因此,为了有效地满足面向互联网的复杂应用对大规模计算能力、海量数据处理和信息服务的需求,将广域分布的异构、自治的资源进行按需组织和管理,以实现动态、跨自治域的资源共享与协同并提高资源综合利用率已成为一个重要的科学问题.

* Supported by the National Natural Science Foundation of China under Grant No.90412011 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2005AA119010 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.2005CB321803 (国家重点基础研究发展计划(973)); the Outstanding Young Research Fund of China under Grant No.60525209 (国家杰出青年基金)

Received 2007-03-15; Accepted 2007-04-26

© 中国科学院软件研究所 http://www.jos.org.cn

为了解决这一问题,需要围绕网络计算模式建立支持资源共享和集成的虚拟计算环境,国际学术界和工业界已开展了大量的研发工作,并取得了积极进展.典型的工作包括:以实现动态、跨自治域网络资源共享协同为主要特征的网格计算(grid computing)、以端到端实现资源和信息服务的对等计算(P2P)、以实现无所不在的计算资源和用户随时随地访问的泛在计算(ubiquitous computing)以及通过整合网络中大量闲散资源为科学计算提供高效计算能力的桌面计算(desktop computing).

近年来,虚拟机技术重新得到重视并得以快速发展,基于虚拟机的虚拟计算环境成为当前网络计算领域的热点研究问题,典型的工作有Virtual Workspaces[1],VIOLIN[2],Virtuoso[3]等等.这些研究项目也从不同侧面展示了虚拟计算环境的共同特点.广义上理解,虚拟计算环境是指“建立在开放的互联网基础设施之上,以网络资源的按需聚合与自主协同为核心机制,为终端用户或应用系统提供和谐、可信、透明的一体化服务环境,实现有效资源共享和便捷合作工作”[4].狭义上理解,虚拟计算环境可视为基于网络的物理资源及其运行管理的支撑软件组成的虚拟组织以及虚拟组织之间的资源共享与协同机制,支持网络资源的有效聚合和可信协同[5].

我们认为,虚拟机(virtual machine,亦称Hypervisor)[6]是在软、硬件之间引入虚拟层,可为应用提供独立的运行环境,屏蔽硬件平台的动态性、分布性和异构性,支持硬件资源的共享和复用,并为每个用户提供属于个人的独立、隔离的计算环境,同时,为管理员提供硬件资源和软件资源的集中管理.然而,基于虚拟机的虚拟计算环境的研发还处于起步阶段,相关工作往往只关注单项虚拟机技术,缺乏与多种虚拟机技术集成方面的考虑;此外,由于缺乏对虚拟计算环境的设计、管理、监控等功能的支持,这些系统无法有效地体现虚拟计算环境的特点.

本文将通过研究分析虚拟机技术和基于虚拟机的虚拟计算环境典型系统,从用户、系统管理员和应用程序3个视角讨论如何基于虚拟机技术有效地构建虚拟计算环境.本文第1节概述虚拟机技术.第2节分析基于虚拟机的虚拟计算环境.第3节给出虚拟计算平台CIVIC(CROWN-based infrastructure for virtual computing)的设计和部分实验分析结果.

1 虚拟机技术

1.1 概 述

在1992年图灵奖颁奖会上,Butler Lampson指出,计算机科学中的任何问题都可以通过在现有系统中引入一个新的中间层的方式加以解决.客观上,分层设计思想是网络和计算机系统的一个主要特点,这种方法使层与层之间相互独立,并按照特定的接口进行通信[7],例如,CPU指令集构成了操作系统和硬件间的接口,系统调用构成了应用程序和操作系统之间的接口.抽象地说,虚拟机技术是通过新增的虚拟中间层截获上层软件对底层接口的调用,并对该调用重新作出解释和处理,以实现异构环境中资源的可共享、可管理和可协同,并支持应用大规模部署、迁移和运行维护.通过虚拟机,可以在原有的硬件资源和操作系统上仿真一台虚拟计算机,使软件可以不经修改直接运行在虚拟机中.

在计算机发展初期,由于操作系统尚未成熟,软件运行对硬件平台具有的独占性和排他性,使共享硬件资源成为当时迫切的需求.1966年,IBM剑桥研究中心设计的VM/360(virtual machine)系统首次在计算机系统中实现了对多用户的支持,为不同用户和应用软件提供独立、共享的运行环境.

随着硬件技术的发展和计算机应用的日益复杂,对软件重用和可移植性的需求不断提高,虚拟机技术逐渐受到重视并不断发展,1980年出现的Smalltalk 80利用虚拟机解释由Smalltalk编译器生成的中间语言,通过虚拟机实现了应用跨平台运行,对应用程序屏蔽了底层硬件平台的异构性,部分地解决了软件的移植和重用问题,实现了“一次编译到处运行”.此后,Sun公司的Java语言也借鉴了虚拟机技术.

由于个人计算机的普及,出现了多种互不兼容的计算机平台,每个平台都包含庞大的应用软件库,软件跨平台性成为迫切的需求.由于Smalltalk类型的高级语言虚拟机并不支持已有应用的跨平台,而是设计新的平台(Smalltalk中间语言)来实现应用的跨平台性,因此缺乏对遗留软件跨平台的支持.1997年,Connectix公司发布的Virtual PC软件[8]支持在Mac机器上运行PC机上的软件,以一种对应用程序透明的方式实现跨平台,使已有程序无须修改即可在虚拟机中运行.

© 中国科学院软件研究所 http://www.jos.org.cn

应该说,早期在个人计算机平台上出现的各种虚拟机主要解决软件的移植性问题,并没有结合虚拟机的原始功能:共享硬件资源和隔离软件运行.1998 年,斯坦福大学创办的VMWare公司发布了第一款在PC机上虚拟PC机的软件VMWare[9].它虽然不能实现软件在不同硬件平台上的可移植性,却可以提供软件隔离和硬件共享能力,在企业获得广泛的应用.

由于硬件平台的升级、维护、失效等状况影响着软件的持续稳定运行,屏蔽硬件资源的动态变化对软件的影响就成为保证软件运行稳定可靠的重要问题.2004 年,VMWare和Xen分别提出了VMotion[10]和Live Migration[11]技术,支持虚拟机的在线迁移,使得基于虚拟机运行的计算机系统可以不间断地从一台物理机器上迁移到另一台物理机器上运行,而迁移过程中系统不可访问时间仅有 100ms左右[11].基于虚拟机在线迁移技术,可以在对上层软件完全透明的情况下,实现硬件资源动态负载均衡,保证软件在硬件设备升级、维护、失效时都能不间断地运行,从而提高软件的可用性和可靠性.

虚拟机的执行效率和稳定性一直是影响虚拟机广泛应用的主要阻力.为了使应用可以不经修改地运行在虚拟机中,虚拟机在最初的实现过程中没有考虑修改操作系统与硬件平台,导致虚拟机实现机制相对复杂.目前,操作系统和硬件平台已普遍对虚拟机加以支持.总体上来说,软、硬件对虚拟机的支持可以有效地提高虚拟机的执行效率和稳定性,并极大地推进了虚拟机的应用.

1.2 主要分类

从静态的角度看,虚拟机是一类系统软件,又称为虚拟机监控器(virtual machine monitor,简称 VMM).虚拟机监控器的核心功能是截获软件对硬件接口的调用,并重新解释为对虚拟硬件的访问;从动态的角度看,虚拟机是一个独立运行的计算机系统,包括操作系统、应用程序和系统当前的运行状态等.因此,虚拟机一词包含两个含义:一个是指虚拟机监控器(VMM 或 Hypervisor),即实现虚拟计算机功能的软件(如 VMWare,virtual PC 等);另一个是指以虚拟机形式运行的系统,如运行在 VMWare 中的 Linux 系统.在本文下面的分析中,为了准确起见,用“虚拟机”一词表示虚拟机监控器,而用“虚拟机实例”一词表示以虚拟机形式运行的系统.

在虚拟机的发展过程中,针对不同的应用需求出现了多种虚拟机,可以按照多种标准给虚拟机进行分类[6,7].下面,我们从虚拟化的规模、指令集、宿主环境等方面进行分析.

(1) 按照虚拟化规模可将虚拟机分为进程虚拟机和系统虚拟机两类.若虚拟机只能为单个系统进程提供虚拟运行环境,则称为进程虚拟机,如 Java,Smalltalk 等高级语言虚拟机;若虚拟机可虚拟整个计算机系统,则称为系统虚拟机,如 VMWare,Virtual PC 等.进程虚拟机比系统虚拟机的运行开销小,实现也相对简单.但系统虚拟机可以用户和应用提供更好的透明性、隔离性、封装性、可管理性.

(2) 按照虚拟机的宿主环境可将虚拟机划分为寄生虚拟机和经典虚拟机.宿主环境是虚拟机的运行支撑环境.如果宿主环境是操作系统,虚拟机作为操作系统的应用程序存在,则称其为寄生虚拟机,如 VMWare 和 Virtual PC;如果宿主环境是硬件平台本身,则称为经典虚拟机,如 IBM 的 VM/360 系统、剑桥大学的 Xen 以及 VMWare 面向企业用户推出的 VMWare ESX Server 等.由于 VMWare 类型的虚拟机拥有大量的用户群,因此寄生虚拟机形式更为人们所熟知,而经典虚拟机可以绕过操作系统直接访问硬件,所以性能上具有较大优势.

(3) 按照宿主机和客户机指令的异同,可将虚拟机分为相同指令虚拟机(same ISA VM)和相异指令虚拟机(different ISA VM).如 VMWare 就是典型的相同指令虚拟机,而早期运行在 Mac 平台上的 Virtual PC 则是相异指令虚拟机.通过相异指令虚拟机可以实现软件在异构硬件资源上的移植性.而相同指令虚拟机则可以直接利用指令相同的特点,直接利用物理平台解释部分虚拟机指令,从而在执行效率上获得一定的优势.

(4) 按照是否需要修改客户机操作系统可将虚拟机分为半虚拟化虚拟机和全虚拟化虚拟机两类.如果虚拟机需要修改客户机操作系统,则称为半虚拟化(para-virtualization)虚拟机;否则,称为全虚拟化(full-virtualization)虚拟机.剑桥大学开发的Xen,User-mode-Linux[12]和OpenVZ[13]都采用半虚拟化技术.全虚拟化具备很好的透明性,即不需要修改操作系统.半虚拟化虽然需要修改操作系统源码,损失了一定的透明性,但对于运行在虚拟机操作系统上的应用程序来说仍然是透明的,而且半虚拟化技术可以降低虚拟机的复杂度.

(5) 按照虚拟机所在中间层位置的不同,可以将虚拟机划分为硬件(HW)虚拟机、操作系统(OS)虚拟机、应

© 中国科学院软件研究所 http://www.jos.org.cn

用程序二进制接口(application binary interface,简称 ABI)虚拟机和应用程序接口(application programming interface,简称 API)虚拟机.

硬件虚拟机在操作系统和底层硬件之间截获CPU指令,如VMWare,Virtual PC[8],Boch[14],Qemu[15]等.操作系统虚拟机位于操作系统和应用程序之间截获操作系统调用,如Linux VServer[16],OpenVZ[13],User-mode-Linux[12]等.ABI虚拟机通过仿真其他操作系统的ABI运行该平台上的应用程序,例如,Wine虚拟机支持在Linux系统中运行Windows程序、FreeBSD系统中的Linux ABI虚拟机支持在FreeBSD中运行Linux的应用程序.

虚拟机也可以对应用程序编程接口(API)进行捕获和解释,如,Cygwin通过仿真POSIX API的方式支持Windows平台上运行Unix下的应用程序.另外,远程桌面技术[17]的实现原理也是截获应用程序对图形用户界面的API的调用进行重新解释,因此也可以认为是一种API虚拟机.总体来说,虚拟机中间层所在的位置越接近硬件层,虚拟机实现的软件隔离性和透明性就越高;中间层位置越接近应用层,虚拟化的粒度就越小,虚拟机的运行所占用的系统开销也就越小.

表 1 总结了上述 5 种分类方法和相关的虚拟机.

Table 1 Category of hypervisors

表 1 常见虚拟机分类

Name	By scale		Is same ISA VM?	By hosted Env		OS transparent		By level indirection
	System VM	Process VM		Classic VM	Hosted VM	Para virtualization	Full virtualization	
IBM VM/360	Y		N	Y			Y	HW
Virtual PC for MAC	Y		N		Y		Y	HW
VMware workstation	Y		Y		Y		Y	HW
VMWare ESX server	Y		Y	Y			Y	HW
QEMU	Y		Y&N		Y		Y	HW
Bochs	Y		Y&N		Y		Y	HW
Xen	Y		Y	Y		Y		HW
Linux Vserver	Y		Y	Y		Y		OS
OpenVZ	Y		Y	Y		Y		OS
User-mode-linux	Y		Y		Y	Y		OS
Wine		Y	Y		Y		Y	ABI
FreeBSD linux ABI		Y	Y		Y		Y	ABI
Cygwin		Y	Y		Y		Y	API
Jails		Y	Y		Y		Y	API
VNC		Y	Y		Y		Y	API

1.3 主要特点

虚拟机可以对软件屏蔽硬件平台的异构性,消除软件对硬件的独占性,实现软件运行的隔离和硬件资源的共享.根据前面对虚拟机应用模式的分析,我们认为虚拟机主要有 4 个特点:

(1) 透明性(transparent):软件可以不经修改地运行在虚拟机中.虚拟机可以为软件提供的运行时隔离、在线迁移等功能都无须修改软件.虚拟机对软件屏蔽了底层硬件平台的异构性,支持软件跨平台运行.

(2) 隔离性(isolation):多个软件可以通过虚拟机互不影响地运行在一台机器上,体现对底层硬件资源的共享.此外,在同一台物理机器上的多个虚拟机实例彼此完全隔离.它们可以各自安装不同版本的软件,而不需要考虑不同虚拟机实例中软件的兼容性问题;还可以互不影响同时运行,而不会因为一个虚拟机实例中的软件失效导致另一个虚拟机实例出错.

(3) 封装性(encapsulation):虚拟机实例中所有软件都封装在一个单独的虚拟硬盘文件中.通过这种封装形式,虚拟机的备份、安装、复制、分发都可以通过复制虚拟硬盘文件的方式实现.有效地降低了软件的管理、配置的难度,增加了软件部署的方便性和灵活性.

(4) 可管理性(manageability):虚拟机的开机、关机、休眠,甚至虚拟硬件的添加、修改、删除等操作都具有编程接口.用户可以通过程序完成对虚拟机的硬件的管理和控制.与手工控制硬件相比,编程控制的方式有着更大的灵活性.将虚拟机编程接口封装成远程服务,可以实现硬件资源的远程管理和集中管理.此外,虚拟机的在线迁移功能同样体现了可管理性,运行的虚拟机实例可以在程序控制下迁移软件运行的硬件平台.

© 中国科学院软件研究所 http://www.jos.org.cn

2 基于虚拟机的虚拟计算环境

由于虚拟机本身具有的透明性、隔离性、封装性和可管理性等特点,基于虚拟机的虚拟计算环境可以结合上述特点,更加有效地整合分散的计算资源,为用户和应用提供一体化的服务环境,实现资源共享和有效利用.对用户来说,基于虚拟机的虚拟计算环境可以为每个用户提供独立于其他用户的运行环境,保证每个用户对其计算环境的独占性.对于计算机管理人员来说,基于虚拟机的虚拟计算环境可以提供计算资源的集中管理功能,使得管理员不必实地操作分散在各处的计算机就可以远程控制硬件和软件的升级和维护.对于应用程序来说,基于虚拟机的虚拟计算环境应该对应用保持透明性,并对应用屏蔽底层硬件资源的动态性、分布性和异构性,保证应用运行的稳定性.

目前,基于虚拟机的虚拟计算环境的典型系统主要分为作业执行环境研究、移动个人计算环境研究和虚拟应用程序研究 3 类:作业执行环境主要面向应用提供一体化的服务环境,包括单台虚拟机的作业执行环境和虚拟机网络两种;移动个人计算环境则面向普通用户提供一体化的服务环境,使得用户可以随时随地访问其个人计算环境;虚拟应用程序面向管理员提供一体化服务环境,使得管理人员可以集中管理整个网络中软件资源的安装、配置和升级.下面将分别介绍上述 3 类典型的研究工作.

2.1 作业执行环境

(1) Globus的Virtual Workspaces项目[1]

Virtual Workspaces 项目的目标是为网格建立可定制和可控的远程作业执行环境.传统的网格主要关注作业的执行,而忽视了作业执行环境的创建、部署和管理.由于作业执行环境的创建需要大量的人工操作来完成软件的安装和配置,加上不同软件之间的依赖性和不兼容性,使得不同作业对其执行环境的配置需求不同,导致不同类型的作业难以共享计算资源.

Virtual Workspaces 项目以单个虚拟机为单位,进行计算资源的分配和管理,可将各种作业执行环境封装在彼此独立的虚拟机实例.由于虚拟机具有封装性,因此,可以通过虚拟机实例的复制完成作业执行环境的部署.利用虚拟机的分割性和隔离性,可以实现多个作业的执行彼此互不影响,共享计算资源.如图 1 所示,客户端可以通过远程接口按需创建虚拟机实例形式存在的网格应用,进行远程配置,用户可以远程启动一个 Virtual Workspace 并开始执行作业,在作业执行完毕后可以通过远程接口及时关闭 Virtual Workspace 并释放其占用的内存和 CPU 资源.

Virtual Workspaces项目为传统的网格运行环境增加了对遗留应用运行管理的支持,支持作业执行环境的无人值守安装,能够有效降低作业执行环境的部署时间,并且降低部署的难度.可以看出,通过虚拟机可以实现网格中CPU、内存、硬盘等计算资源的分配和调度[18],为不同类型的作业分配CPU个数、内存大小、硬盘和网络带宽的配额.在实现硬件资源共享的同时,还可以保证每个计算任务对计算资源的独占性和运行的隔离性.

但在实际使用中,有一些计算任务需要运行在独立的网络环境中,而 Virtual Workspaces 不支持这类运行环境.

(2) 普渡大学的VIOLIN项目[2]

VIOLIN 为需要运行在独立的网络环境中的应用提供具有网络拓扑结构的虚拟网络运行环境.用户可以通过 VIOLIN 为不同的应用设计网络拓扑,形成由多台虚拟机组成的隔离的虚拟网络环境.

图 2 是VIOLIN的体系结构图.位于途中第 2 层的覆盖网基础设施是VIOLIN系统的运行环境.通过覆盖网基础设施在物理网络上建立的虚拟网络环境称为一个VIOLIN.通过VIOLIN,用户可以在任意的物理网络拓扑结构下按需创建逻辑的虚拟网络拓扑结构,还可以定制虚拟网络中每一台虚拟主机和虚拟网络设备的软件配置和硬件配置.例如,利用VIOLIN可以在不支持IP多播的物理网络上架设虚拟IP多播网络,并在虚拟网络上运行支持IP多播功能的视频点播软件.这种方式既不需要修改视频点播软件,也不需要设计一种新的应用层多播的协议.此外,利用VIOLIN可以创建虚拟移动IP网络测试移动IP协议栈的稳定性和性能指标,在无须实际物理设备和终端实际移动的前提下实现对真实协议栈功能的测试,降低了测试环境的部署成本和测试过程的人工参与度,

© 中国科学院软件研究所 http://www.jos.org.cn

可以此为基础实现在不同网络拓扑下网络软件的自动测试[19].

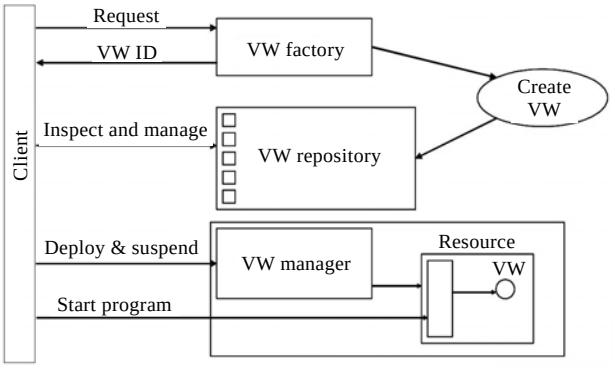

Fig.1 Virtual Workspaces interaction

图 1 Virtual Workspaces 交互过程

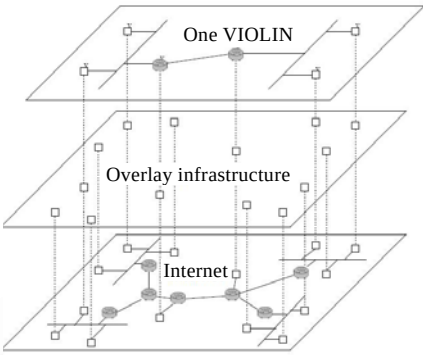

Fig.2 VIOLIN architecture

图 2 VIOLIN 体系结构

然而,目前虚拟机网络项目仅关注虚拟机网络的运行支持,而缺乏对虚拟机网络的设计、部署、修改等静态层面的支持,用户不能方便地定制虚拟机网络.

2.2 移动个人计算环境

Intel的Internet Suspend and Resume(ISR)[20]目的是实现个人计算环境的移动性.与移动计算的终端移动性不同,ISR不需要用户携带任何便携式的移动计算设备,而是将用户个人计算环境运行在虚拟机中,通过虚拟机的移动实现个人计算环境的移动.

如图3所示,ISR中虚拟机实例的移动是通过分布式文件系统来完成的.ISR需要将虚拟机实例和虚拟机休眠文件保存在分布式文件系统上.如果用户需要在任何一台计算机上恢复其个人计算环境,只需通过Internet将分布式文件系统中的相关文件复制到本地即可恢复虚拟机实例的运行;用户关闭或者休眠虚拟机之后,需要将本地虚拟机实例文件同步到服务器上.由于只是虚拟机发生了移动,用户无须随身携带任何硬件设备,随时随地打开任何一台可以访问Internet的计算机,就可以恢复他的个人计算环境.

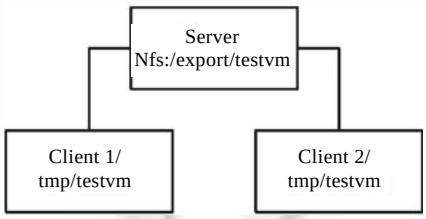

Fig.3 ISR test scenario

图 3 ISR 测试场景

ISR 的缺点在于,需要用户手动完成休眠和恢复等操作,且有一段时间计算机处于停机状态.如果将虚拟机的在线迁移功能与ISR相结合,就可以实现移动计算环境的不间断运行,提供一直在线的移动计算环境.但是,当前的在线迁移功能限制在局域网的机器之间,无法实现Internet上的迁移,限制了计算环境的移动范围.

2.3 虚拟应用程序

Softricity公司的SoftGrid产品[21]可以实现软件的集中管理.在计算机发展早期,软件都是由计算机管理员统一管理,随着个人电脑的普及,计算机用户也同时身兼管理员的角色.由于各种病毒、木马和其他恶意软件导致软件的频繁更新、升级,增加了软件维护上的困难,同时使得计算机的运营维护成本一直居高不下.

SoftGrid 可以在不修改现有软件编程模型和运行方式的前提下,通过虚拟机技术来实现软件的集中管理.这种技术又称为虚拟应用程序(virtual application,简称VA).如图4所示,SoftGrid由服务器端系统和客户端系统构成.服务器端的打包模块(preparer)负责将应用程序转化成VA格式,所有的VA都集中存放在服务器的软件库(VA repository)中.而用户计算机上需要安装SoftGrid的客户端系统,用户在运行任何应用程序之前,需要首先调用传输模块(deliverer),将所需的应用程序的一部分下载到本地软件缓存(VA cache)中,再由执行模块(executor)执行.执行模块是一种进程虚拟机,为每个VA提供虚拟的执行环境.如果用户已经把VA下载到了本机的缓存中,则执行模块可以直接运行VA,提高执行效率.

© 中国科学院软件研究所 http://www.jos.org.cn

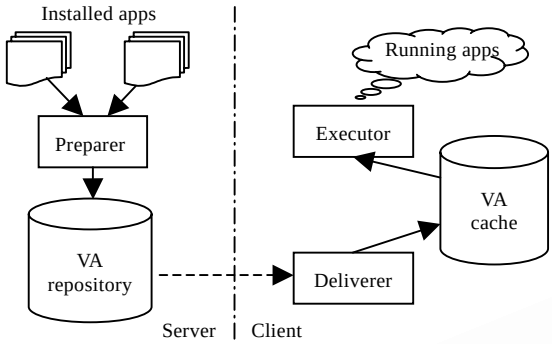

Fig.4 SoftGrid system architecture

图 4 SoftGrid 体系结构

VA下载后无须安装即可直接由客户端运行,而SoftGrid的打包模块可以将Windows应用程序自动转化成VA,包括Office办公软件.通过类似SoftGrid的服务端系统,如PDS[22],可以实现软件的集中管理.原先一个应用程序的更新涉及每台计算机上的软件更新,而通过虚拟应用程序技术,则只需将新的应用程序打包并更新服务器软件库,由此实现自动管理.

2.4 小 结

本节分别从应用程序、用户和管理员 3 个角度分析了作业执行环境、移动个人计算环境和虚拟应用程序的研究进展,体现了虚拟计算环境应该具备的一致的服务环境的特点.其中,Virtual Workspaces 提供了整合计算机资源的能力,向用户屏蔽了底层计算环境的异构性,简化了作业执行环境的管理和部署;而 VIOLIN 则可以按需提供多台虚拟机构成的网络,保证不同用户虚拟环境的隔离性;ISR 可以实现移动的虚拟计算环境,使用户可以随时随地用任何设备访问属于自己的个人环境;而 SoftGrid 系统可以提供软件的集中管理,使得管理员可以像管理一台计算机那样管理由 Internet 上多台计算机组成的虚拟计算环境,从而降低了管理员的管理难度.

然而,这些系统往往只关注单一的虚拟机技术,在系统体系结构的设计时没有综合考虑与各种类型的虚拟机的集成,因此,往往只能体现虚拟计算环境的一部分特点,难以针对单一系统进行扩展以支持更多的虚拟计算的特性,将互联网真正变成一台虚拟计算机.此外,这些系统主要关注实现虚拟计算环境中的运行时支持,缺乏对静态层面的支持,也缺乏按照用户的需求来设计、构建和配置所需的虚拟机实例和虚拟机网络的支持.

3 CROWN 虚拟计算平台

针对上述问题,我们设计了CROWN虚拟计算平台CIVIC.它是服务网格系统CROWN[23]3.0 版本的核心组件,可以为CROWN网格平台中的网格服务提供独立、隔离的执行环境,有效地屏蔽底层运行平台的异构性、分布性和动态性,并提供虚拟计算环境的管理监控功能. CIVIC中集成了对多种虚拟机技术的支持,可以体现虚拟计算环境多方面的特点.当然,多种虚拟机的引入将增加系统复杂度,并在一定程度上影响效率,但却可以有效地提升网格服务执行的稳定性和可靠性.

3.1 系统设计

虚拟计算平台 CIVIC 的设计目标是力图较完整地实现基于虚拟机的虚拟计算环境,包括对计算机资源的共享与协同、对应用程序提供透明的支持以及对用户和管理员提供集中管理视图.

CIVIC 系统包含运行视图和管理视图(如图 5 所示).

运行视图共分为 4 层,自下而上依次是:

(1) 资源层(resource layer):由分布在 Internet 上动态、异构的计算机组成,资源层的节点称为资源节点.

(2) 容器层(container layer):容器层中的节点是安装了 CIVIC 容器软件的硬件资源,称为容器节点(container node).通过 CIVIC 容器可以实现硬件资源的虚拟化,具体方式是每个容器节点上都可以运行多台虚拟机实例.

© 中国科学院软件研究所 http://www.jos.org.cn

容器节点提供远程调用接口,通过这些接口可以对运行在该节点上的虚拟机实例进行控制,如远程部署、启动、停止等操作.

(3) 协调层(coordination layer):协调层中的节点是特殊的容器节点,称为协调节点(coordinator node),协调节点是配置了协调功能的容器节点.协调节点负责管理其他容器节点,在它们之间形成了一个覆盖网.在协调层,针对不同目的有不同的覆盖网,比如,资源管理覆盖网用于虚拟资源的注册和查询,虚拟机网络覆盖网在多个容器节点之间维护稳虚拟机网络的稳定运行等.

(4) 实例层(instance layer):实例层的节点 CIVIC 为用户提供作业的运行环境,称为实例节点(instance node). CIVIC 共支持 3 种不同的实例节点:虚拟机实例、虚拟机网络(VM network,简称 VMN)和虚拟应用程序(VA).其中,虚拟机实例中包含完整的操作系统,虚拟应用程序中只包含单个应用;用户可以通过多种交互方式访问实例节点,如访问实例节点的远程图形用户界面,或以网格作业提交的方式运行实例节点中的应用.

CIVIC 管理视图提供 3 类管理工具:辅助用户创建所需的计算环境的设计单元(CIVIC designer),可对容器节点、协调节点、实例节点进行管理和监控的管理单元(manager)以及监控单元(monitor).

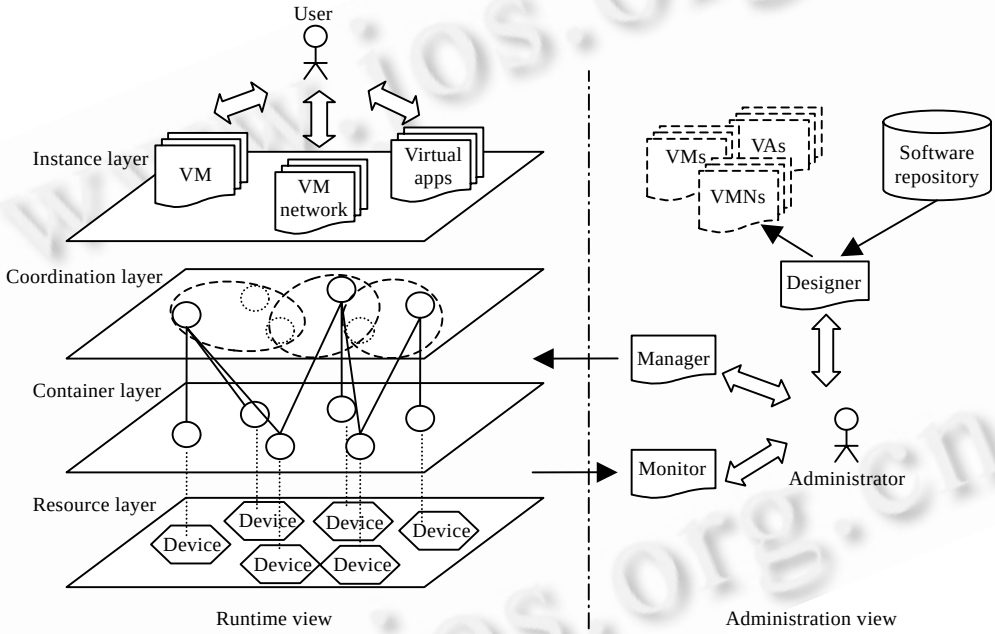

Fig.5 CIVIC system views

图 5 CIVIC 系统视图

3.2 模块功能设计

如图 6 所示,CIVIC 模块功能包括资源层、容器层、协调层、实例层和交互层 5 层.

(1) 资源层:包括各种硬件资源、存储资源等,这些计算资源构成了 CIVIC 系统中底层资源.

(2) 容器层:位于资源层上层,又可以细分为虚拟机监控器子层和远程控制子层:

(a) 虚拟机监控器子层:包含系统虚拟机、进程虚拟机和远程界面 3 个模块.该层直接采用当前成熟的虚拟机技术(如 Xen,VMWare),安装了系统虚拟机的容器节点可运行多个虚拟机实例;进程虚拟机可以使用 Java 等,安装了进程虚拟机的机器将支持运行虚拟应用程序;远程界面模块使用 VNC,该模块将用于支持用户与实例节点的交互.

(b) 远程控制子层:为容器节点提供远程管理服务.该子层包含虚拟机管理接口、网络管理接口和信息查询接口等模块.通过虚拟机管理接口可以对运行在容器节点中的虚拟机实例进行热部署、启动、关闭、休眠、恢复和在线迁移等远程操作.通过网络管理接口管理容器节点的网络连接,利用隧道技术连接多个容器节点,可以

© 中国科学院软件研究所 http://www.jos.org.cn

在容器节点之间形成由多个虚拟机实例组成的虚拟网络.信息查询接口为上层监控程序提供节点基本信息的查询功能,包括对物理机器的硬件资源,如 CPU、内存、硬盘使用情况的查询,以及运行在容器节点上多个虚拟机实例的硬件资源使用情况的查询.

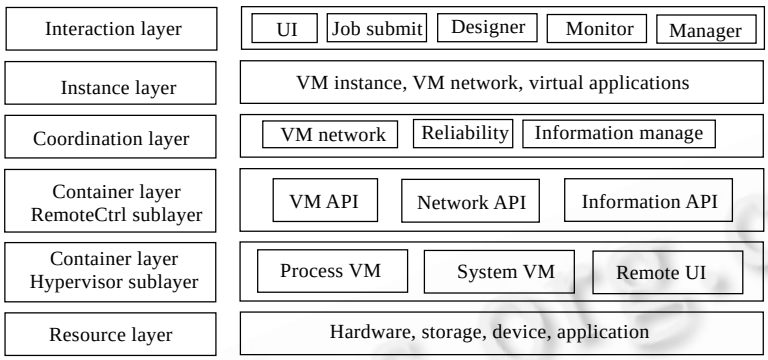

Fig.6 CIVIC system modules

图 6 CIVIC 系统模块

(3) 协调层:提供协调节点的各种维护功能模块,其中包含虚拟机网络维护、可靠性维护、资源拓扑维护等模块.前面提到协调节点是特殊的容器节点,如安装了虚拟机网络维护模块的协调节点可以在多个容器节点上创建多台虚拟机,并调用容器节点的网络管理接口把虚拟机连接成虚拟机网络;同时,还负责在容器节点发生动态变化时,利用虚拟机的在线迁移功能维护虚拟机网络的持续运行.

安装了可靠性维护模块的协调节点可以实时监控虚拟机实例的运行状况,并在检测到虚拟机实例失效时,将虚拟机恢复到之前保存的运行状态下,从而保证虚拟机实例的稳定运行.协调节点通过资源拓扑管理模块监控多个节点和虚拟机实例的运行状态,多个协调节点之间将形成资源层叠网.

(4) 实例层:该层对应用程序保持透明性.透明性主要体现在 3 个方面:首先,虚拟机可以屏蔽底层硬件资源的异构性对软件的影响,软件通过虚拟机可不经修改地运行在不同的硬件平台上.通过虚拟机的在线迁移功能,还可以屏蔽底层硬件资源的动态性和分布性,软件可以在运行过程中迁移底层运行的硬件平台;其次,透明性体现在遗留软件可以运行在 CIVIC 中而不需要做任何修改;最后,通过虚拟机的失效检测和重配置服务,可以增加应用程序运行的可靠性和安全性.

(5) 交互层:包含两类交互模块.一类交互模块提供用户对实例节点界面访问的支持,例如,通过远程图形界面访问虚拟机实例,或者访问虚拟机实例的命令行界面.用户可以通过容器节点提供的作业提交 Web 服务接口调用虚拟机实例中的应用程序.另一类交互模块为用户提供虚拟机实例的设计、管理、监控支持.设计单元提供一体化的可视化编辑器来辅助用户按需创建计算环境,同时支持以拖拽的方式创建具有特定拓扑结构的虚拟机网络.

3.3 虚拟机实验

CIVIC 系统正在开发中.目前,CIVIC 已支持虚拟机实例和虚拟机网络的设计和部署功能(如图 7 所示).

通过对 CIVIC 系统的虚拟机实例和虚拟机网络的创建、部署、启动等功能的测试,得到如表 2 所示的实验结果.实验所用的环境是两台 CPU 为 Intel P4 3GHz,内存为 1G 的实验用机,其中一台用于虚拟机实例和虚拟机网络的创建,另一台作为运行环境.实验结果中记录了不同类型的实例的创建时间、部署时间、启动时间和生成实例文件的大小.

该实验中的创建的虚拟机实例是一个最小化的 Linux 系统,安装了基本的网络程序如 ftp,telnet 等,一个虚拟机实例实际占用 136 兆字节磁盘空间.从表 2 的实验结果可以看出,通过 CIVIC 设计单元创建一台全新的 Linux 虚拟机实例只需要大约 7 分钟;如果把已有的虚拟机实例作为模板创建新实例,则可以将创建时间减少为

© 中国科学院软件研究所 http://www.jos.org.cn

2 分钟左右,且这段时间无须人工参与.一般情况下,系统安装一台物理机器的时间大约在 30 分钟~60 分钟之间,而且安装过程中需要人工参与.相比之下,可以发现 CIVIC 能够有效地缩短应用程序的安装配置时间.而且,从表中实验结果可以看出,CIVIC 只需 2 分钟左右就可以自动完成 3 台虚拟机实例组成虚拟机网络的部署过程,而在实际中,多台物理机器的安装和组网所需要的时间远不止这个数量级.由此可见,基于网络的应用可以在 CIVIC 系统中能够得到迅速、有效的部署.

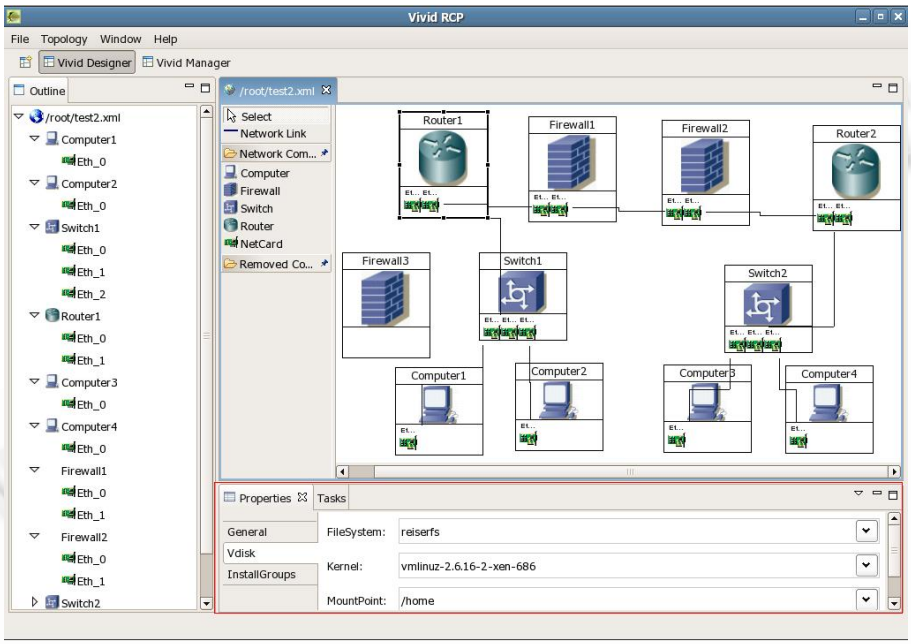

Fig.7 CIVIC designer

图 7 CIVIC 设计单元

Table 2 VM instance install experiments

表 2 虚拟机实例的创建实验

Type of instance	Size (MB)	Install time (s)	Deploy time (s)	Boot time (s)
One VM instance (fresh)	136	441	45	79
One VM instance (template)	136	143	45	79
One VM network with two VM instances	273	246	88	94
One VM network with three VM instances	409	351	129	113

4 结束语

本文首先介绍虚拟机技术的出现和发展,通过对虚拟机技术典型应用的分析,总结出虚拟机具有透明性、隔离性、封装性和异构性 4 个特点.然后,通过对虚拟机的相关研究分析,给出了基于虚拟机的虚拟计算环境的 3 个特点:可以为每个用户提供属于个人的独立、隔离的计算环境;可以为计算机管理人员提供硬件资源和软件资源的集中管理功能;同时对应用程序保持透明性,并屏蔽底层硬件资源的动态性、分布性和异构性.文中分析了 Virtual Workspaces 和 VIOLIN 等 3 类基于虚拟机的虚拟计算环境的典型系统,最后给出了 CROWN 虚拟计算平台 CIVIC 的系统功能的设计和实验分析.

References:

[1] Keahey K, Foster I, Freeman T, Zhang XH, Galron D. Virtual workspaces in the grid. In: Cunha JC, Medeiros PD, eds. Proc. of the 11th Int'l Euro-Par Conf. Springer-Verlag, 2005.

[2] Ruth P, Jiang XX, Xu DY, Goasguen S. Virtual distributed environments in a shared infrastructure. IEEE Computer, 2005,38(5):

© 中国科学院软件研究所 http://www.jos.org.cn

39−47.
[3] Virtuoso: Resource management and prediction for distributed computing using virtual machines. 2007. http://virtuoso.cs.northwestern.edu/
[4] Lu XC, Wang HM, Wang J. Internet-Based virtual computing environment (iVCE): Concepts and architecture. Science in China (Series E), 2006,36(10):1081−1099 (in Chinese with English abstract).
[5] Rousselle P, Tymann P, Hariri S, Fox G. The virtual computing environment. In: Proc. of the 3rd IEEE Int'l Symp. on High Performance Distributed Computing. 1994. 7−14.
[6] Rosenblum M, Garfinkel T. Virtual machine monitors: Current technology and future trends. IEEE Computer, 2005,38(5):39−47.
[7] Smith JE, Nair R. The architecture of virtual machines. IEEE Computer, 2005,38(5):32−38.
[8] Microsoft virtual PC 2007. 2007. http://www.microsoft.com/windows/virtualpc/default.mspx
[9] VMware—Virtualization software. 2007. http://www.vmware.com/
[10] Nelson M, Lim BH, Hutchins G. Fast transparent migration for virtual machines. In: Proc. of the USENIX Annual Technical Conf. 2005. 391−394.
[11] Clark C, Fraser K, Hand S, Hansen JG, Jul E, Limpach C, Pratt I, Warfield A. Live migration of virtual machines. In: Proc. of the 2nd ACM/USENIX Symp. on Networked Systems Design and Implementation (NSDI). 2005. 273−286.
[12] Dike J. A user-mode port of the linux kernel. In: Proc. of the 4th Annual Linux Showcase Conf. (ALS 2000). 2000. 63−72.
[13] Server virtualization open source project—OpenVZ. 2007. http://openvz.org/
[14] Bochs the open source IA-32 emulator. 2007. http://bochs.sourceforge.net/
[15] Open source processor emulator—QEMU. 2007. http://fabrice.bellard.free.fr/qemu/
[16] Linux-VServer project. 2007. http://linux-vserver.org/
[17] Baratto RA, Kim LN, Nieh J. THINC: A virtual display architecture for thin-client computing. In: Proc. of the ACM SIGOPS Operating Systems Review.New York,: ACM Press, 2005. 277−290.
[18] Figueiredo R, Dinda PA, Fortes J. Resource virtualization renaissance. IEEE Computer, 2005,38(5):28−31.
[19] Bavier A, Bowman M, Chun B, Culler D, Karlin S, Muir S, Peterson L, Roscoe T, Spalink T, Wawrzoniak M. Operating system support for planetary-scale network services. In: Proc. of the 1st Symp. on Networked Systems Design and Implementation. 2004. 253−266.
[20] Kozuch M, Satyanarayanan M. Internet suspend/resume. In: Proc. of the Workshop on Mobile Computing Systems and Applications. IEEE Computer Society, 2002. 40−46.
[21] Greschler D, Mangan T. Networking lessons in delivering ‘software as a service’. Int’l Journal of Network Management, 2002, 12(5):317−321.
[22] Alpern B, Auerbach J, Bala V, Frauenhofer T, Mummert T, Pigott M. PDS: A virtual execution environment for software deployment. In: Proc. of the Virtual Execution Environments. New York: ACM Press, 2005. 175−185.
[23] Huai JP, Hu CH, Li JX, Sun HL, Wo TY. CROWN: A service grid middleware with trust management mechanism. Science in China (Series E), 2006,36(10):1127−1155 (in Chinese with English abstract).

附中文参考文献:

[4] 卢锡城,王怀民,王戟.虚拟计算环境 iVCE:概念与体系结构.中国科学(E 辑),2006,36(10):1081−1099.
[23] 怀进鹏,胡春明,李建欣,孙海龙,沃天宇.CROWN:面向服务的网格中间件系统与信任管理.中国科学(E 辑),2006,36(10):1127−1155.

怀进鹏(1962－),男,黑龙江哈尔滨人,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机软件与理论,网络安全,网格计算.

胡春明(1977－),男,博士,讲师,CCF 会员,主要研究领域为网格技术,网格 QoS.

李沁(1982－),男,博士生,主要研究领域为分布式计算,网格技术.

© 中国科学院软件研究所 http://www.jos.org.cn