面向远程虚拟桌面的应用程序推送技术研究

蒋媛园[1] 武延军[2]
(中国科学院软件研究所 北京 100190)[1] (中国科学院研究生院 北京 100190)[2]

摘 要 远程虚拟桌面是用户桌面使用环境的虚拟化,可实现对操作系统及应用程序的集中管理和高效分发、迁移,使得用户在具备基本的硬件条件下使用自己的工作环境。虚拟桌面应用程序推送方案RVDvApp是在图形指令传输过程中实现过滤机制,推送单独的应用程序到客户端,从而实现基于虚拟桌面的服务分发,使得用户可以克服异构的执行环境,获取集中部署在服务器端虚拟机上的应用程序服务。

关键词 远程虚拟桌面,图形指令过滤,应用程序推送
中图法分类号 TP316.8 **文献标识码** A

Remote Virtual Desktop Oriented Application Pushing Research

JIANG Yuan-yuan[1] WU Yan-jun[2]
(Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)[1]
(Graduate University of Chinese Academy of Sciences, Beijing 100190, China)[2]

Abstract Remote virtual desktop is the virtualization of desktop working environment, which can achieve centralized management, efficiently distribute and migrate operating systems and applications. The motivation is to help users to use their own working environment with the basic condition of the hardware. A solution of pushing applications of virtual desktop, named RVDvApp, was proposed. The core mechanism of RVDvApp is to implement a filter in the process of graphic commands transmission. As a result, the separate application can be pushed to the client and the virtual desktop based services can also be distributed. Therefore, users can totally ignore the heterogeneous working environment and get application services centrically deployed on the server-side virtual machine.

Keywords Remote virtual desktop, Graphic commands filtering, Application pushing

1 简介

虚拟桌面是典型的云计算应用,它能够在"云"中为用户提供远程的计算机桌面服务。服务提供者在数据中心服务器上运行用户所需的操作系统和应用软件,然后用桌面显示协议将操作系统桌面视图以图像的方式传送到用户端设备上。同时,服务器对用户端的输入进行处理,并随时更新桌面视图的内容。

远程虚拟桌面是用户使用环境的虚拟化,它强调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条件就可以使用自己的工作环境。实现技术上有VNC、RDP等远程桌面技术。面向普适终端的云计算技术研究也日渐成为热门[1]。

本文提出的面向远程虚拟桌面的应用程序推送技术RVDvApp是一种新的应用程序虚拟化解决方案。该方案在远程虚拟桌面图形指令的传输过程实现图形指令过滤机制,仅提取应用程序图像,将应用服务推送到客户端,客户端无需进行完整桌面的渲染,让用户在虚拟使用环境下获得应用程序虚拟化的体验。面向远程虚拟桌面的应用程序推送的实现,使得应用程序虚拟化框架更加高效。

2 背景和相关工作

2.1 Spice简介

Spice(Simple Protocol for Independent Computing Environments)是红帽公司维护的虚拟桌面开源项目,为实现云环境下的独立计算环境提供高性能通信协议[2]。Spice协议是一种专门设计应用于虚拟环境的自适应远程传送协议。Spice旨在利用现代的高性能通信技术为今天的带宽密集型应用(如多媒体、VoIP)提供无缝的用户体验。

Spice实现开放的远程计算方案、客户端远程接入机器设备、虚拟机环境。Spice在服务器端为每个用户准备其专用的虚拟机并在其中部署用户所需的操作系统和各种应用,然后通过桌面显示协议将完整的虚拟机桌面交付给远程用户。

Spice协议为访问、控制、远程接入设备、发送输出等功能定义了一组协议消息[3]。不同于使用帧缓冲更新方法的其它

到稿日期:2012-07-15 返修日期:2012-11-04 本文受"核高基"国家科技重大专项(2010ZX01036-001-002, 2010ZX01037-001-002, 2010ZX01036-001-002-2),中国科学院知识创新工程重要方向项目(KGCX2-YW-174)资助。

蒋媛园(1988—),女,硕士生,主要研究方向为操作系统技术,E-mail:jiangqxq@gmail.com;武延军(1979—),男,博士,副研究员,硕士生导师,主要研究方向为操作系统。

远程桌面方案，Spice 可在客户端进行原始图像的绘制。

2.2 相关工作

本文面向远程虚拟桌面的应用程序推送方案 RVDvApp 是远程计算环境下应用虚拟化的一种形式。在远程计算环境下，用户访问一个服务器虚拟化后的应用时，用户计算机只需要把交互操作通过协议传送到服务器端，应用程序的计算逻辑在服务器端为用户开设的会话空间中运行，把更新后的人机交互逻辑传送给客户端，并在客户端用相应设备展示出来。目前成熟的技术有 Go-Global、SeamlessRDP、Xen-App。

Go-Global 是一组虚拟化应用交付平台套件，它把服务端作为应用程序服务器，发布各类操作系统的应用程序[4]，用户远程登录后即可使用安装在服务器的应用。但 Go-Global 采用的 RXP 协议非开放，并且没有借助虚拟化实现服务器端硬件资源的扩展性。

SeamlessRDP 是 Cendio 软件公司的开源项目，提供了针对 rdesktop 的无缝窗口实现方案[5]。SeamlessRDP 仅支持 Linux 的客户端，不能满足异构环境下用户的需求。

XenApp 是 Citrix 公司的非开源技术，采用 ICA 协议实现的按需应用交付解决方案，能够在数据中心对所有 Windows 应用实现集中管理[6]。但 XenApp 不支持特定的应用推送，并且 ICA 协议不开放、不开源，普通用户需要支付较高的费用。

本文设计实现的 RVDvApp 方案实现了应用程序的独立推送，并遵循开放的协议。对于 RVDvApp 的访问协议，我们基于 Spice 进行了改造，从而相比于基于 RDP 和 ICA 等协议实现的上述应用虚拟化技术，RVDvApp 方案支持服务器端硬件资源的可扩展，支持不同操作系统的客户端，用户可以克服异构的环境，获取服务器的应用程序服务；并且要求更低的带宽，提供流畅的用户交互体验。在广域网环境下，Spice 比 RDP 有更好的性能[7]，其次对比于 RDP 和 ICA 协议，它支持访问客户端硬件，比如打印机、U 盘，所以 RVDvApp 方案更适于云中心和多客户端一体化的实现。

远程计算环境实现的应用虚拟化有别于传统的应用层虚拟化技术，如 VMware 的 ThinApp[8]、Microsoft 的 MS App-V 是对应用程序安装过程进行打包，然后进行分发部署，每一个程序有一个相对独立的依赖环境，这就必然会有依赖环境数据的冗余；对应用程序使用过程打包实现的应用虚拟化，如 CDE[9] 目前是半自动化的打包工具，对于依赖环境数据的完备性有较高的要求。若用户只需在客户端访问推送的应用程序，对于一个应用程序来说，不存在运行依赖环境数据完备性要求的技术瓶颈；对于应用程序之间，也不存在依赖环境数据的冗余问题；并且自动化程度较高。

2.3 方案的提出

综合对现有技术的分析，云计算环境中应用推送解决方案要求服务器端的资源部署有良好的可扩展性，也要求客户端能支持异构的环境。本文提出的 RVDvApp 方案，就是针对虚拟环境实现的应用程序推送方案。整体架构中，虚拟机在服务器端进行有效的部署，应用服务再在虚拟机上进行部署，用户可在不同操作系统下的客户端通过应用推送方案，以更低的带宽要求使用虚拟运行环境，这将比传统方式有更高效的资源利用。用户即可在虚拟使用环境下获得应用程序虚拟化的体验。RVDvApp 结合虚拟桌面协议特性，实现应用程序的图像过滤机制，远程推送应用程序到客户端，旨在实现远程虚拟计算环境下应用虚拟化的一种新的解决方案。结合 Spice 协议，该方案的整体系统结构如图 1 所示。

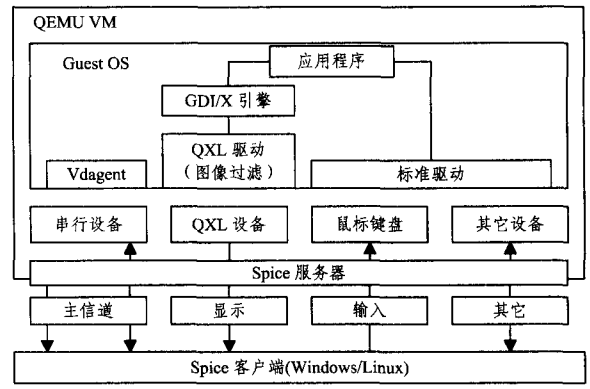

图 1 系统结构图

RVDvApp 推送方案在 Spice 协议的架构上实现了创新，它的核心机制是在 Spice 协议虚拟桌面推送流程的关键步骤——QXL 虚拟图像设备驱动工作层实现图像过滤，分离出桌面和应用程序的图像，将应用程序推送到客户端。下面以核心过滤层设计基础、设计、过滤数据粒度和过滤实例的思路，阐述实现虚拟运行环境中应用程序推送的 RVDvApp 方案。

如图 1 所示，Spice 协议服务器端的 Guest OS 运行在虚拟环境中，完全虚拟化能够为虚拟机中的操作系统提供一个与物理硬件完全相同的虚拟硬件环境，不需要修改操作系统代码，虚拟环境支持多种操作系统的运行部署。Spice 直接与虚拟化的环境连接，而传统的远程显示协议是与虚拟机中运行的操作系统连接。客户端支持 Windows 和 Linux 环境。客户端和服务器通过信道通信，处理程序能够根据不同的信道类型对消息进行具体处理。

RVDvApp 应用推送方案，借助虚拟桌面进行服务的分发，用户在虚拟的使用环境下获得应用虚拟化的体验。当本地运行环境受限或者授权受限而无法安装某些软件时，用户可以连接远程服务，服务器只需要把使用的应用程序的图像推送到客户端，而不必显示完整的桌面。该方案的实现将会对虚拟应用环境的共享实现方案提供支持。在此基础上，如果将不同客户端的输入事件独立，使多客户端共享一个虚拟机，那么还可以减轻服务器运行虚拟机的负载，增强整个系统运行的性能。

3 设计和实现

3.1 设计原理

Qemu 针对 Spice 增加了 QXL 图形显示设备等虚拟设备，Spice 服务器作为 Qemu 的一部分也运行在宿主机上。Spice 服务器与远程的客户端使用 Spice 协议进行通信，同时也与 Qemu 进行交互。

Spice 服务器可以传输图形指令到客户端，由客户端的 GPU 来完成渲染。显示通道连接之后，客户端发送初始化消息给服务器端。服务器给客户端发送显示模式消息，指定当前区域尺寸和格式，客户端收到消息后创建绘图区域，接下来

根据后面的命令对这个区域进行渲染。Spice 协议里封装了图形命令的格式，并且在 Spice 的图形设备 QXL 会把传统的图形命令转换成符合传输要求的命令格式。Spice 图形命令流如图 2 所示。

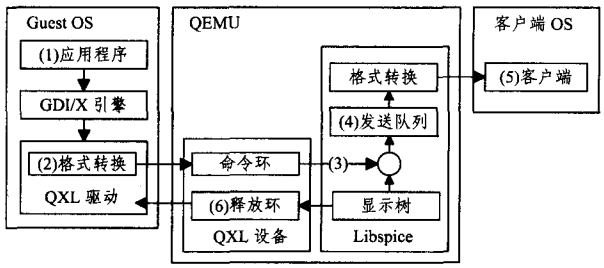

图 2　Spice 图形命令流

每个应用程序对应自己的画图区域，应用程序在被分配的区域里进行图像的绘制，例如文字显示、图片显示，甚至是原型图像的绘制。通常，服务器端应用程序的一部分源位图资源会经过预渲染，再随渲染命令传输到客户端，根据命令要求进行合成操作，以显示不同的图像效果。所以可以根据为应用程序分配的位图资源和操作区域对图形指令进行过滤。

3.2　RVDvApp 逻辑架构

结合 Spice 的实现机制，本文提出面向远程虚拟桌面的应用程序推送方案 RVDvApp 的逻辑架构，如图 3 所示。在 QXL 虚拟设备的驱动程序中加入图像过滤层。在驱动程序处理过程中，应用程序和桌面的图像绘制是分别完成然后合成显示，在过滤层的作用下即可分离出应用程序的图形指令并推送到客户端进行渲染。

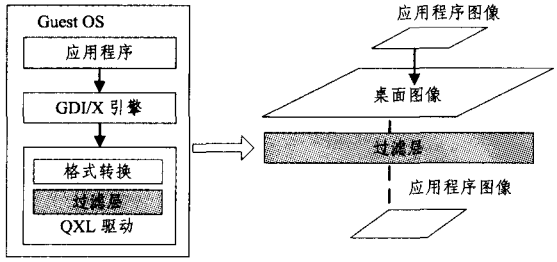

图 3　RVDvApp 逻辑架构图

3.3　核心过滤层

3.3.1　过滤层设计基础

驱动程序可把某个应用程序的图像绘制在离屏表面(offscreen surface)上，它实际对应一部分内存区域。把离屏表面作为绘图源表面的特性称为 Surface 特性。对于 Win7 来说，这个性能可以让 Windows 为每一个顶层的窗口建立一个表面，这样既能减少重画的工作，也有利于不同应用程序图像的分离。对应于离屏表面，所有的绘图工作基于基础表面(primary surface)，它直接对应于显示器，并根据显示器的分辨率进行自适应调整。

所有的绘图命令扩展一个目标表面的字段，指明在哪个 surface 上绘图。Spice 服务器的代码也被扩展，可以支持多个表面，每个表面使用一个树结构，在树里记录不同表面的依赖。

每个应用程序可以维护自己的 Surface 信息，使应用程序图像与桌面图像分离，亦即使 RVDvApp 核心过滤机制的实现成为可能。

3.3.2　过滤层设计

Qemu 针对 Spice 增加了虚拟图像显示设备 QXL，Spice 服务器支持 QXL VDI，当 Qemu 使用虚拟设备接口库 LibSpice 时，Qemu 用 QXL PCI 设备来加强远程显示性能和 guest OS 的图形处理能力。QXL 驱动是图形指令处理的关键功能模块，RVDvApp 的实现需要着重分析 QXL 驱动已有的功能和关键的工作流程，在合适的位置插入过滤层。

QXL 驱动分为两个功能模块：首先是显示驱动功能模块，它是一个内核模式的 DLL，主要负责渲染；其次是 miniport 驱动功能模块，它处于内核模式，通常处理必须与内核组件交互的操作，负责资源管理[10,11]。以 guest OS 为 Windows 为例，显示设备驱动程序与图像引擎的交互如图 4 所示。

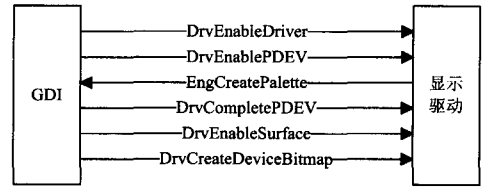

图 4　显示设备驱动程序与图像引擎的交互图

图形指令真正的传输和客户端绘制工作在最后两个步骤进行。绘图和文本都依赖于一个绘制的表面。DrvEnableSurface 创建基础表面，它的信息对应表面信息的链表头，另外，此函数里还要进行其它 SurfaceInfo 的初始化工作。

之后每次应用程序有界面显示请求，都会创建新的离屏表面，DrvCreateDeviceBitmap 实现这一功能，并分配专有的 surface_id。驱动程序负责确定各种表面所支持的绘图操作的类型。

由此可见，主表面的正确绘制是所有其他图像能够正常显示的前提，它的初始化数据不能被过滤拦截。应用程序开始申请自己的离屏表面是过滤机制实现的关键位置，在此可以实现数据的分流。

3.3.3　过滤数据粒度

QXL 驱动会把图形的相关指令转换为 Spice 定义的格式，压入 QXL 设备的命令环并传输到客户端。QXL 绘图命令的结构维护特定字段，指明绘图操作都是针对一个表面来进行的。但表面并不一一对应一张位图，每个表面有多个位图引用，包括各种具备不同功能的资源，每一个资源都可以是一个 GDI 位图或者设备相关位图。RVDvApp 的过滤机制在每一次命令发送前判断它关联的表面信息。

3.3.4　绘图命令过滤实例

应用程序有界面显示请求，都会有新的离屏表面被创建。DrvCreateDeviceBitmap 可实现这一功能，每次新创建表面时获取当前进程信息，给表面信息扩展新的字段来存储进程镜像文件名。通常一个应用程序都会创建多个 Surafce，它们共同合成最后图像的显示。在驱动向设备压入命令之前，通过表面的字段数据获取表面信息，即判断目标应用程序是否需要在客户端绘图。

但是 Spice 不支持绝对的客户端渲染，有些位图是在服务器端预渲染后拷贝到客户端。并且所有的绘图和文字显示都是基于主表面，它被各个应用进程共享。属于某个应用程

序的预渲染图像就有可能直接绘制到主表面，而不在自己维护的表面信息范围内。所以在被绘制的表面是主表面的情况下，当前程序若不是需要在客户端显示的应用程序，它的图像指令应该被过滤掉，不进行网络传输。

4 实验与评价

4.1 实验环境

QXL 驱动编译环境：安装支持 Windows7 的 WinDDK 驱动编译开发环境，拷贝 Spice 协议源码包到编译工作目录，并设置环境变量指明头文件的路径，其中包含有关 QXL 数据结构定义的头文件。

Spice 实验环境基于 Fedora16(64-bit)搭建 Spice 的服务器和客户端。服务器端 Host 机安装 Qemu，guest OS 选择 Windows7，Windows7 安装 QXL 驱动、虚拟 IO 串口的驱动程序以及 vd-agent 程序。

驱动程序相对于应用层程序需要更加复杂的调试环境，所以环境的搭建和使用都有一定难度和复杂度。调试环境搭建基于 Qemu 用于调试 guest OS 的驱动程序的 WinDbg 调试环境[12]。为了在内核模式调试 guest OS，需要搭建一个安装 WinDbg 程序的虚拟机作为调试机，并把 Spice 服务器端的 guest OS 作为被调试的目标机。Qemu 提供的重定向功能支持在同一个物理机上“远程”连接调试机和目标机。根据 WinDbg 提供的机制，调试环境启动后，等待目标机的连接申请，目标机以调试模式启动后与调试机连接，WinDbg 程序根据设置的路径参数，以远程复制的方式更新目标机的图形驱动程序，驱动程序正常启动后，WinDbg 可以跟踪程序的运行并输出调试信息。

4.2 实验效果

对比原来整个桌面的推送，RVDvApp 方案可以仅把应用程序的图像传输到 Spice 客户端并渲染，以并发的 3 个应用程序(画图，记事本，计算器)为例说明实验效果，如图 5 所示。

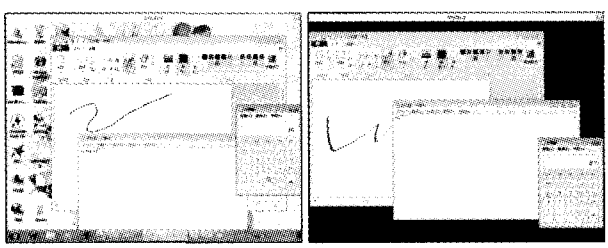

图 5 实验效果图

4.3 实验评估

4.3.1 网络传输数据

为了对比推送某个应用程序和推送整个桌面的数据传输量，本文的测试选用 IPTraf 网络监控工具[13]来监测服务器和客户端交互过程的网络负载。通过 IPTraf 设置特定的网络端口，监测服务器和客户端有网络数据传输行为的进程，包括服务器的 Mosaicsyss 进程以及客户端的 Stdptc、Xnds、Gdbremote、Touchnetpl、Trp 进程。

应用程序的推送选择了 Office Word、IE 浏览器、画图 3 款软件；统计了它们运行过程中，服务器和客户端交互数据的平均值和完整虚拟桌面显示情况下的交互数据量对比，如表 1 所列。

表 1 数据量对比

源进程	目的进程	完整桌面(Bytes)	应用程序(Bytes)
Mosaicsysssptc	Stdptc	259445	258861
Stdptc	Mosaicsyss	5572	5716
Brdptc	Mosaicsyss	102046	25732
Brdptc	Mosaicsyss	496	496
Xnds	Mosaicsyss	108914	8372
Gdbremote	Mosaicsyss	546385	219373
Mosaicsyss	Xnds	1827596	171680
Touchnetpl	Mosaicsyss	322942	167686
Mosaicsyss	Gdbremote	51084632	5841349
Mosaicsyss	Brdptc	480143	120184
Mosaicsyss	Trp	654	654
Mosaicsyss	Touchnetpl	172558	88290

在仅推送应用程序的情况下，数据平均传输量为完整桌面推送情况下的 52.03%。

4.3.2 用户体验实时性

图像的过滤在 DisplayChannel 对应的虚拟设备和驱动中完成，所以服务器端和客户端相连接的其它信道没有受到影响，例如 InputsChannel 正常接收客户端的键盘和鼠标输入。关于程序的响应，客户端使用通过图像过滤出来的单个应用程序与在完整的虚拟桌面上使用并没有明显的差异。

结束语 本文提出的 RVDvApp 方案在 Spice 协议架构上实现了创新，在 Surface 特性、QXL 设备驱动程序等关键技术基础上，以核心过滤层设计基础、设计、过滤数据粒度和过滤实例的思路，阐述了在虚拟运行环境中实现应用程序推送的 RVDvApp 方案的设计和实现。该方案在客户端只显示需要的应用程序界面而不是整个桌面，用户可以根据自己的需要远程获取软件服务。在本文实现的技术支持下，用户可以克服异构的客户端运行环境，获取集中部署在服务器虚拟机上的应用程序服务。

RVDvApp 方案实现了服务器虚拟机到客户端一对一的应用程序推送。在此基础上，研究多客户端接入服务器的共享虚拟运行环境技术，以减轻服务器运行虚拟机的负载。

参 考 文 献

[1] 陈援非，崔丽，朱珍民，等. UbiCloud：一种面向普适终端的云计算系统[J]. 计算机科学，2011，38(10)：127-132

[2] Spice：spice_for_newbies[EB/OL]. http://Spice-space.org/

[3] Spice：spice_protocol[EB/OL]. http://Spice-space.org/

[4] Go-Global：Go-Global for Windows[EB/OL]. http://www.graphon.com/products-and-solutions/

[5] SeamlessRDP：ThinLinc Administrator's Guide for ThinLinc [EB/OL]. http://www.cendio.com/seamlessrdp/

[6] XenApp：XenApp6.5 简介[EB/OL]. http://www.citrix.com.cn/products/xenapp/

[7] Mikhail K，Solomatina B，et al. Benchmark Report on SPICE Remote Display Protocol[R]. NCSD of Russia Test Report. 2010

[8] Thin App：VMware ThinApp User's Guide[EB/OL]. http://www.vmware.com/pdf/

[9] Guo P J. CDE：Run Any Linux Application On-Demand Without Installation[C]//USENIX LISA. Boston，MA，2011

[10] 张帆，史彩成. Windows 驱动开发技术详解[M]. 北京：电子工业出版社，2008：1-120

[11] Cant C. Windows WDM 设备驱动程序开发指南[M]. 孙义，译. 北京：机械工业出版社，2000：1-150

[12] KVM：QEMU_user_manual[EB/OL]. http://www.linuxkvm.org/page/Documents

[13] Iptraf：Police Your Network Traffic with IPTraf [EB/OL]. http://iptraf.seul.org/