清 华 大 学

综 合 论 文 训 练

题目：云计算虚拟机的安全访问与共享

系　　别：自动化系

专　　业：自动化

姓　　名：钱瀚

指导教师：曹军威　研究员

2010 年 6 月 29 日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签　名：戎瀚　导师签名：唐宇成　日　期：2010.7.1

中文摘要

虚拟化是云计算的基石，虚拟资源的访问与共享是云计算的基础。虚拟社区的成员可以共享自己的虚拟机，为其他成员提供计算、存储和服务资源，以此来实现廉价、高效、系统的资源利用。

上述目标实现的一个落脚点是对虚拟机的访问和虚拟机的共享控制。本文先讨论使用 RDP 协议进行远程访问，在此基础上通过 Java Applet 加载应用程序来实现基于 web 浏览器的远程虚拟机访问。

接着，本文对虚拟机的共享机制作出探讨，设计虚拟机安全共享的体系结构。在这个结构中将采用数字证书验证身份，通过 VPN 来建立安全的连接，通过服务器的端口控制策略来控制虚拟机访问的权限，从而实现虚拟社区内虚拟机的安全共享。

关键词：云计算；虚拟机；web 浏览器；OpenVPN

ABSTRACT

Virtualization is the cornerstone of cloud computing, and the access and sharing of virtual resources are the bases of cloud computing. Members of Virtual Organization can share their own virtual machines, which provides others computing, storage, and service resources in order to achieve cheap, efficient, and systemic resource utilization.

To achieve the goal above, we should pay attention to the access and sharing control of virtual machines. In this paper, we first analyze the RDP protocol for remote access, based on this the web browser load an application to remote access virtual machines by using Java Applet.

Then we discuss the sharing mechanism of virtual machines, and come up with the architecture to achieve secure sharing. It authenticates users using digital certificates, establishes secure connections through the VPN, and control access by using port control strategy at the server.

Keywords：cloud computing; virtual machine; web browser; OpenVPN

目 录

第1章 引 言

云计算作为2009年度最受关注的十大IT技术之一，其商业模式和计算模式受到产业界和学术界的广泛关注。云计算的构想是使用户通过互联网随时甚至随地获得近乎无限的计算能力和丰富多样的信息服务。虚拟化是实现这一构想的基石，它的伸缩性和灵活性可以方便资源利用、简化服务的管理和维护。

虚拟资源的访问与共享是虚拟化过程中具体而基本的问题，通过对虚拟机的安全控制用户可以使用基本的虚拟化应用。本文主要就对于虚拟机的安全访问和共享做出探讨，提出基于浏览器访问的初步架构，并在此基础上使用安全机制来保证资源共享的安全性。

本章将从云计算的虚拟化出发，论述虚拟资源访问与共享的背景和意义。然后讲述本课题的研究任务和目标，最后给出本论文的组织结构。

1.1 研究背景

1.1.1 云计算发展概述

云计算通过将各种互联的计算、存储、数据、应用等资源进行有效整合并实现多层次的虚拟化与抽象，有效地将大规模的计算资源以可靠服务的模式提供给用户，用户则不用考虑复杂的底层硬件逻辑、网络协议、软件架构等问题[1]。用户通常不需要知道云内部的结构和技术实现，仿佛置身云端，这个云端就是远程的分布式系统。这些分布在各地的系统通过互联网集合在一起，使用开放的技术和标准，实现对硬件和软件的动态配置和扩展，这样用户就能享受到强大数据集群支持的服务。一个案例可以说明上述过程：

IBM公司的研究院分布在世界各地，其中某些实验需要海量的存储空间和计算资源，某些实验需要各研究院协同完成。出于这两方面的考虑，IBM公司构建了IBM Research Compute Cloud (RC2) 将分散在各个研究院的资源（包括服务器和存储）整合提供给内部实验人员使用，保证了实验所需硬件支持；同时该系统可以记录保存实验的整个过程，既保证了数据的安全，又方便各研究院实验人员的随时访问和交换。

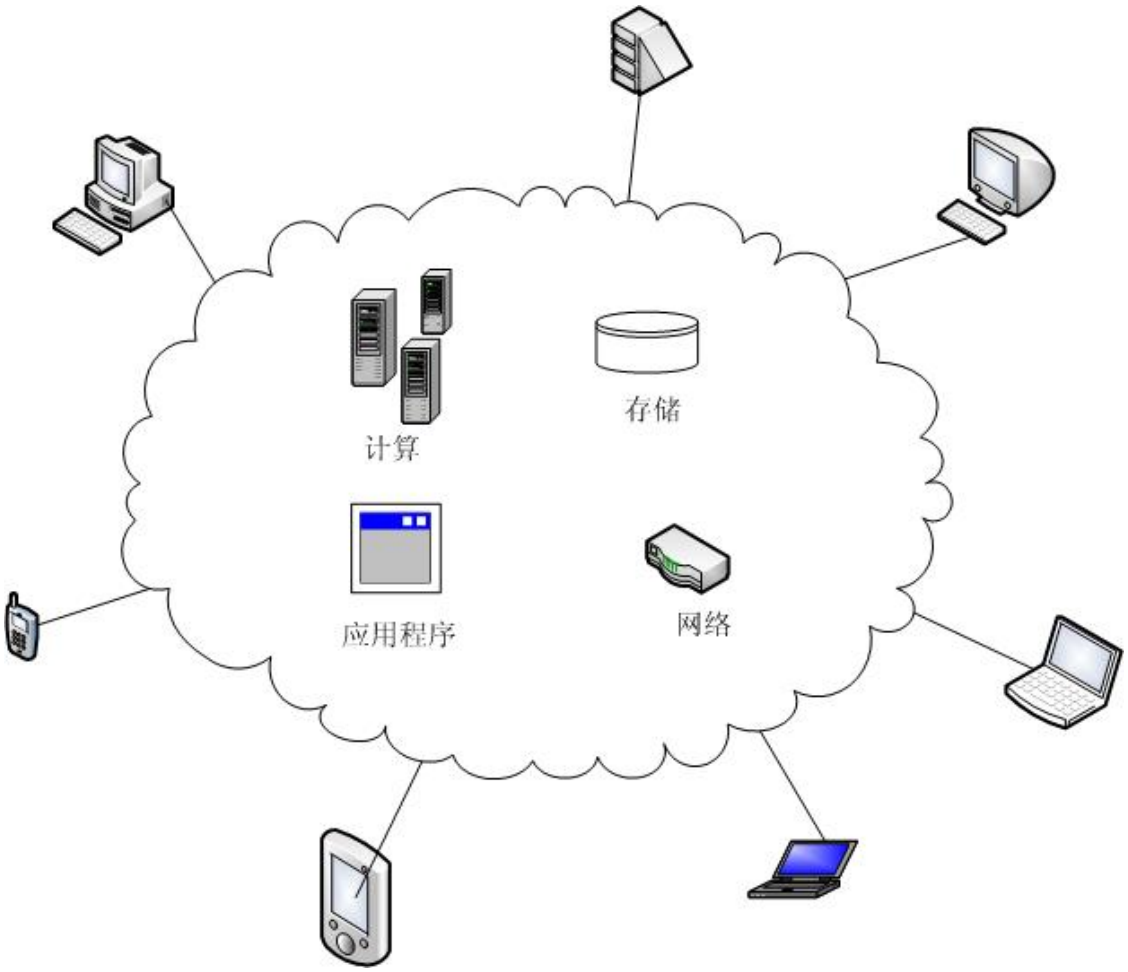

图 1.1 简单云平台

在上述案例中，实际上实验人员不需要管理和维护这些资源。而 IBM 公司努力整合资源的结果，不仅降低了运营成本，同时大大加速了科研的进程。这就是云计算中资源整合与共享的意义所在。与上述案例不同的是，更多的云是为用户提供服务的第三方公用平台。目前全球有数家公司提出云计算基础设施或云计算平台，有的已经投入商业使用：Google 云计算基础设施[2]、IBM“蓝云(blue cloud)”计算平台、Sun 云基础设施、微软 Azure 云平台、Amazon 弹性计算云。图 1.1 显示了简单云计算平台的“云”示例，云端支持各种外设通过有线或者无线接入网络，真正做到随时随地使用云平台。

云计算有四个关键的特征[3]：

一、 云平台中的硬件和软件都是资源，用户可以享有硬件支持和软件服务；

二、 云平台中的资源都可以根据需要进行动态扩展和配置；

三、 云平台中的资源在物理上以独立分布式存在，但在逻辑上以单一整体的形式呈现给用户；

四、 用户按需使用资源，按量付费但不需管理这些资源。

1.1.2 资源虚拟化

虚拟化是实现云计算构想的重要基础。“云”这样一个超级计算机存储了所有的数据、应用程序等资源，面对着众多种类的硬件设备，如何统一地管理这些硬件资源变得尤为重要。虚拟化技术为此提出了很好的解决方案。虚拟化技术可以将物理资源等底层架构进行抽象，使得设备的差异和兼容性对上层应用透明，从而允许云对底层千差万别的资源进行统一管理[3]。虚拟化过程中形成相当数量的虚拟机，每台虚拟机保留相应的应用和服务，这样有效地形成隔离，一个应用的崩溃不至于影响到其他虚拟机的应用。同时，使用虚拟化可以方便地对资源进行调度实现按需分配。此外，虚拟机易于创建和移植，这就保证了应用和服务的可靠性和可用性。

目前主要有四家厂商致力于虚拟化开发：IBM 公司的优势在于大型机和小型机的高端虚拟化，同时其虚拟化领域的产品解决方案可以帮助数据中心实现虚拟化布署、监控和管理自动化；VMware 公司占据着 x86 虚拟化市场的较大份额，他们主要工作一直是数据中心服务器的虚拟化；Citrix 公司在收购 Xen 后成功主导着 x86 服务器虚拟化，他们的服务包括服务器虚拟化、应用虚拟化和桌面虚拟化，“将数据中心变为交付中心”是他们的战略构想；Microsoft 公司在软件研发方面的强大实力为他们在 x86 虚拟化市场上开辟了一片市场，他们的产品线比较全面，尤其是虚拟化管理服务。

虚拟化技术的大力发展推动了云计算的推广。应用虚拟化，云平台中的计算、存储、应用都成为可以动态扩展和配置的资源。云计算将这些资源作为整体推向用户，用户可以获得高效、可靠、廉价、系统的服务。

1.1.3 桌面虚拟化

最近几年，虚拟化市场的重心在服务器虚拟化方面，而桌面虚拟化未受到重点关注。对于此，思杰(Citrix)公司副总裁兼大中华区总经理曹衡康的观点是：“今年(2010 年)会是桌面虚拟化的元年，桌面虚拟化市场会出现明显快速的增长。”[4]

桌面虚拟化是指支持实现桌面系统的远程动态访问与数据中心统一托管的技术，通过这个技术，我们可以通过任何设备，在任何地点，任何时间访问在网络

上的属于我们个人的桌面系统。“云计算代表了人们对未来 IT 体系架构的诉求：每个人，不管使用什么终端和平台，都不会被禁锢住。”思杰(Citrix) 桌面虚拟化部产品市场副总裁 Sumit Dhawan 这样强调桌面虚拟化作为“云计算的灵魂”的重要意义[5]。

实施桌面虚拟化对于企业来讲具有不可低估的现实意义，除了帮助企业降低终端维护成本和节能降耗外，还可以保护企业的知识产品安全。用户的信息和对应的应用系统集中保存在服务器中，采用统一的终端信息保存与维护，防止数据分散在用户层面，可以使企业摆脱病毒和黑客的滋扰。另一方面，服务器统一管理便于系统的统一升级和补丁，也增加了安全性，同时提高工作效率。

在上述企业桌面虚拟化环境中，实现安全性的前提是用户可以有效地安全地接入服务器获取应用服务。有效性指服务器对于多平台终端的适应能力，由于这种架构将所有的计算集中到服务器进行，这样终端设备就可以不仅限于传统的计算机，也可以是笔记本、PDA 甚至手机等也可以实现类似于个人计算的体验；安全性指服务器拥有基于角色的访问控制和安全接入连接的功能。

1.2 课题任务

本课题主要研究个人用户对虚拟资源的利用。如第 1 节所述，服务器保存用户的信息和应用系统，实施桌面虚拟化的首要任务就是与服务器建立有效而安全的连接。课题分两部分：虚拟机的访问与虚拟机的共享。

虚拟资源在服务器以虚拟机的形式存在，通过对远程控制虚拟机从而获得存储、计算、应用等服务，用户与虚拟机建立有效的连接是云计算中最基础的问题。本课题作为构建虚拟社区的一部分，必须保证结果简洁、高效。图 1.2 是虚拟机访问的示意图，Server 作为服务提供端，可能由数台或更多服务器组成，每台服务器提供若干数目的虚拟机。用户在获得虚拟机的使用权限后，根据路径与之建立连接，就可以操控该虚拟机，进行计算、存储或者应用。一个用户可以同时使用多台虚拟机，但为了保证可靠性，一台虚拟机在任意时刻最多只能与一位用户建立连接。使用 web 浏览器作为访问工具，是考虑到客户端的多样性和易用性：客户端的操作系统可能是 Windows / Linux / Mac，终端可能是笔记本/PDA/手机等，浏览器在各个系统之间是通用的，而且不需安装，所以使用 web 浏览器极大的保证了虚拟机访问的操控性。

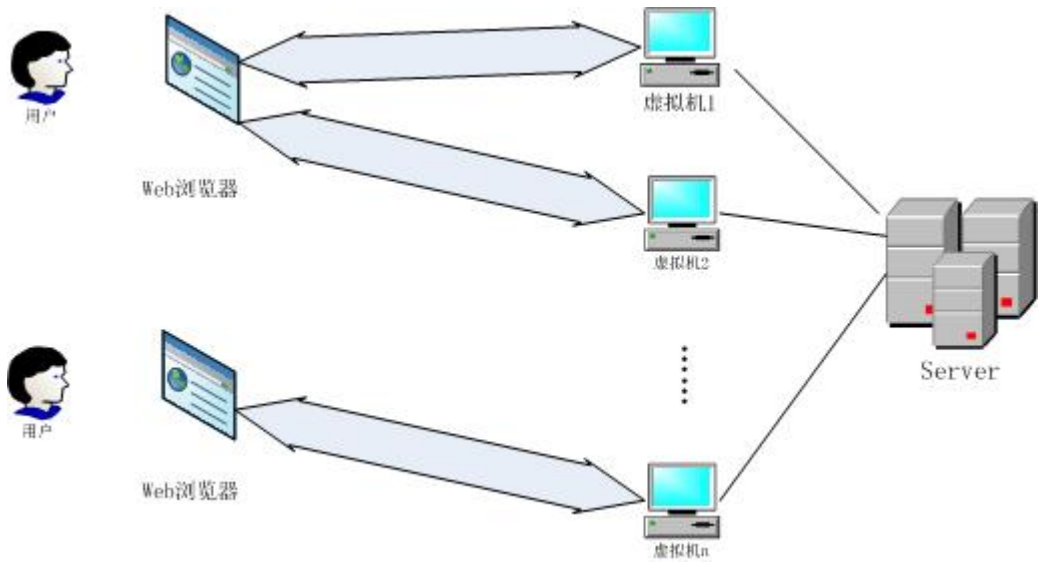

图 1.2 浏览器访问虚拟机

共享是虚拟社区里的概念。虚拟社区可以形成有共同兴趣或者工作相关的用户群，在这个群内实现资源的共享利用。具体地讲，某用户在申请获得认证之后加入某 Group，该 Group 的其他成员可以向其开放自己虚拟机的使用权限，同组内成员可以实现相互之间的虚拟机共享，以此节省资源配置减少工作量。

具体的研究任务分为三个阶段：虚拟机的安装与配置、基于 web 浏览器的访问、认证共享。各阶段说明如下：

1) 虚拟机的安装与配置

考察 x86 的几款主流虚拟机软件 VMware 和 VirtualBox，重点考察该软件在 Linux 系统下的易用性和安全机制，易用性包括方便复制、移植、无图像界面下的操作等，安全机制包括虚拟机的访问控制、网络连接安全等。然后在 Linux 系统下配置环境变量和网络连接，将虚拟机配置为方便共享的状态。

2) 基于 web 浏览器的访问

考察几款远程访问软件的多平台支持和可扩展性，本课题所需的访问必须支持多个平台，包括 Windows、Linux 及 Linux 的扩展系统，可扩展性指对于浏览器的支持，考察对象定为 VNC 和 Rdesktop。另外，基于浏览器的访问需要 Java 的支持，本课题将其定为 html 与 Java Applet 的协同开发。

3) 认证共享

用户在获得某虚拟机的访问权限后会得到认证中心授权的证书，通过证书来建立与虚拟机的连接通道，这就需要编写基于 openssl 的证书认证程序。同时，考虑到同一服务器里众多虚拟机的访问控制，认证过程必须配合防火墙的使用。

1.3 论文结构

本论文主要针对虚拟机的访问和共享进行研究，从而实现在客户端对虚拟资源的方便、可靠的使用。

论文第 1 章是引言，介绍了课题背景和研究任务，并对任务的实现做了简要的技术说明。

论文第 2 章考察几款 x86 的虚拟机软件，根据课题需求做出选择，安装完毕后进行相应的配置。

论文第 3 章介绍基于 web 浏览器访问虚拟机的实现目标，考察几款远程访问软件并做出选择。使用 Java Applet 帮助实现浏览器加载应用程序，从而达到使用浏览器方便地访问远程虚拟机的目的。

论文第 4 章介绍虚拟机安全共享的实现方式，讨论使用 OpenVPN 实现虚拟专用网络的搭建，从而实现对访问权限的控制和传送数据的加密。

论文第 5 章对整个论文做简要总结，对毕业设计做自我评价，并对后续的工作提出继续研究和努力的方向。

1.4 小结

本章首先概述了云计算的发展，和云计算对资源虚拟化的需求，阐明课题研究的背景意义。然后对具体的研究任务作出描述，并将其分为三个阶段——虚拟机的安装与配置、基于 web 浏览器的访问、认证共享，并对各个阶段的任务做了简要的说明。最后给出了论文的结构组织。

第2章 虚拟机的安装与配置

2.1 虚拟机概述

不同于 Java 那样提供介于硬件和编译程序之间的软件，本文所指的虚拟机都是指由软件虚拟出来具有模拟真实的特定硬件环境的计算机。虚拟机是虚拟化进程的产物。虚拟化最早可以追溯到 20 世纪 60 年代，IBM 公司作为虚拟化技术最早的推动者发明了一种操作系统的虚拟机技术，它能在一台主机上运行多个操作系统，使得当时昂贵的大型机资源得到充分利用，这被认为是虚拟化进程中里程碑式的事件。

随着计算机硬件的发展与性能的飞速提高，虽然能够在一个硬盘上创建多个主分区，并且在每个主分区上安装不同的操作系统，但毋庸置疑的是采用这种方法一台计算机在同一时刻只能运行一个操作系统。此时，英特尔和 AMD 公司对于 x86 的研发和对 x86 架构的虚拟化技术的支持，使得虚拟化开始普及，人们对于廉价、高性能可靠的服务器的追求成为虚拟化发展的强劲动力。

1999 年 VMware 公司推出了针对 x86 系统的虚拟化技术，发布了 VMware Workstation 1.0 软件，这个软件在一台正在运行的计算机上模拟出一个虚拟的计算机硬件环境，在这个环境上可以运行另外一个操作系统（包括 Windows、DOS、NetWare、Linux 等），这开启了 x86 系统的个人虚拟机新时代。之后 VMware 推出 server 级别的虚拟软件，为企业级的用户提供了虚拟化解决方案。VMware 公司无疑为个人及企业虚拟化进程起到了巨大的推动作用，该公司目前也占据着市场较大的份额。根据 VMware 的资料，目前世界财富 500 强企业中有 95%在使用 VMware 的产品，其号召力可见一斑。

微软依靠收购 Connetix 进入虚拟化市场，旗下的产品包括不断更新的 Virtual PC 虚拟机软件。在服务器方面，从最初的 Virtual Server 到 2008 年发布的经典作品 Windows Server 2008，完成了质的跨越。Windows Server 2008 中加入了虚拟化技术——Hyper-V。Hyper-V 采用微内核架构，用于连接硬件和虚拟机。Hyper-V 兼顾了安全性和性能的要求，代码非常少，所以代码执行效率更高更可靠；此外 Hyper-V 中不包含任何第三方的驱动，使得系统更加精简。从架构上将 Hyper-V 使服务器能够充分利用硬件资源，使虚拟机系统性能更接近真实系统性能。按照

微软公司提出的“提供统一虚拟化解决方案”的战略构想，Hyper-V 提供了从桌面虚拟化、应用虚拟化、服务器虚拟化道展现层虚拟化的完整生产线。

以上两个公司在企业服务器级别的虚拟化方面有着成熟的技术、完善的产品和稳固的市场。在桌面虚拟化领域，还有一支正在崛起的力量，就是本文的主角——Sun 公司推出的 VirtualBox。VirtualBox 是一款功能强大的 x86 虚拟机软件，不仅具有丰富的特色，而且性能很优异。更为关键的是 VirtualBox 开源，是发布在 GPL (General Public License) 许可之下的自由软件，这无疑为其他自由软件的开发提供了非常大的便利。

虚拟机并不是只是运行在一台计算机上的多个操作系统，其主要功效如下：

1) 减少能源消耗，提高服务器利用率。随着计算机硬件性能的增强，耗电量也随之增加，而随着服务器数量的增加，本身及配套设施的耗电量也以几何级数增长。另一方面，许多服务器的利用率却偏低，造成资源的浪费。采用虚拟机技术能保证更高效的使用。
2) 整合服务器，提高运营效率。虚拟机能帮助在单个服务器上实现多个应用程序互不干扰地运行，以此来整合服务器，从而减少硬件需求、降低功耗、降低冷却和空间面积需求。
3) 优化软件开发和测试。检验软件开发成果，不用担心操作系统的崩溃，链接和复制虚拟机的便利也简化了多层应用程序的开发和测试过程。

总之，虚拟机技术的发展可以帮助个人用户和服务器充分利用计算机资源，能够减少管理成本与使用成本，并且加速软件开发等实验进程。

2.2 虚拟机的选择

2.2.1 VMware 产品

考虑到课题的背景和实现，我们首先需要对虚拟机软件做出取舍。VMware 因其强大的功能、完善的产品体系和强势的市场占有率成为我们的首选，目前 VMware 的一套虚拟化产品线包括：面向大型企业的 VMware ESX Server、面向中小企业的 VMware Server 和面向个人用户的 VMware Workstation。

VMware ESX Server 是一个适用于任何系统环境的企业级虚拟机软件，它运行在一个定制的 Linux 主机系统上，所以软件本身不需要主机操作系统的支持，可以直接运行在硬件层。其大型机级别的架构提供了空前的可用性和操作控制，完全动态的资源控制，适合各种要求严格的应用程序的需要[6]。

另外一款 Server 级别的软件 VMware Server 是面向“工作组”的部门级虚拟机产品，在一台物理主机上同时运行 64 台虚拟机，这样一台服务器可以同时作为多台服务器使用。VMware Server 的一个特点是不需要登录进入系统，在主机启动后就可以自动运行指定的虚拟机，这极大地方便了服务器的运行。另外，同 VMware ESX Server 一样，VMware Server 提供远程 web 管理或者其他控制台管理功能。此外，VMware Server 还提供“一次快照”的保存与还原能力，这些功能都保证了服务器管理的方便和可靠。图 2.1 展示了 VMware Server 通过浏览器启动虚拟机管理界面，输入 https://VMware server:8833 即可启动。令人遗憾的是，该软件的 web access 是基于 apache tomcat 的，占用比较大的资源，对于虚拟机的管理来说有些得不偿失。

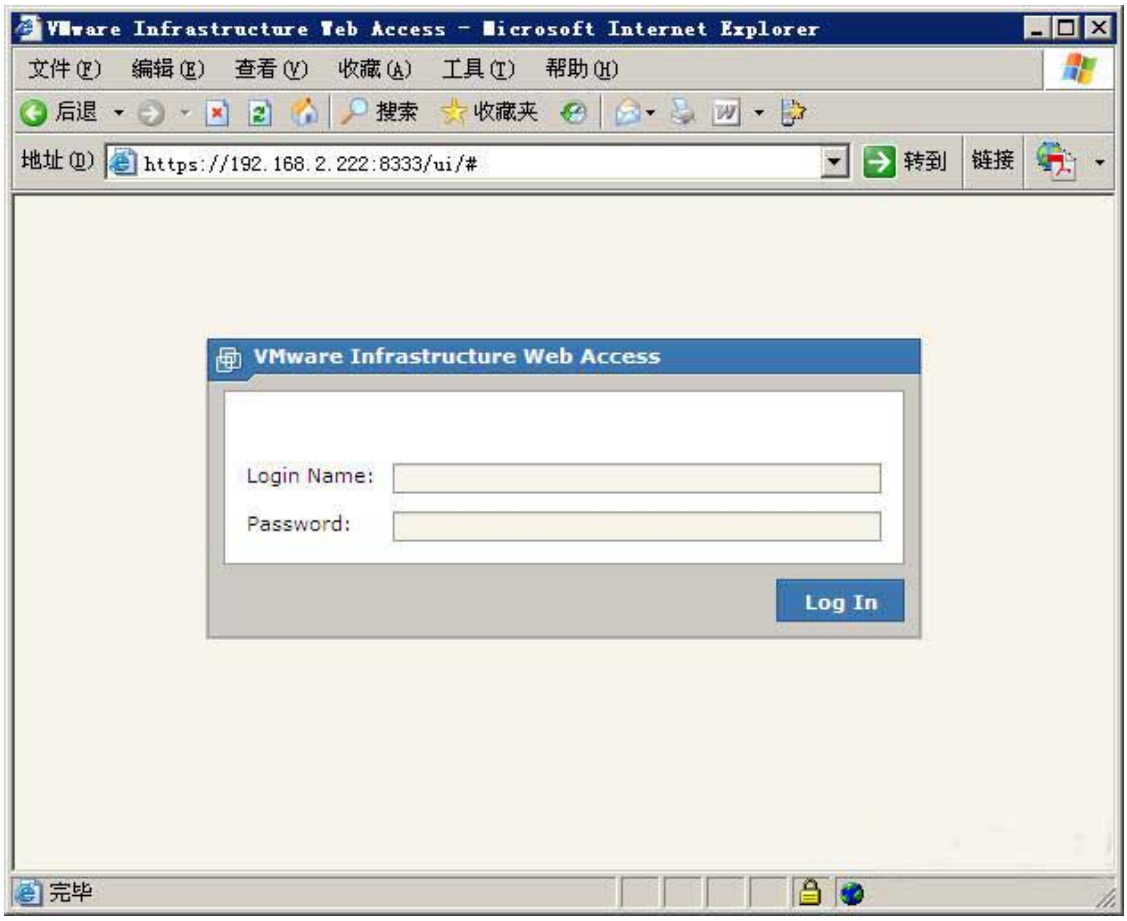

图 2.1 VMware 的 web access 界面

还有一款在个人用户中普及率很高的虚拟机软件——VMware Workstation。它能在本地提供性能优秀的虚拟服务器，但是不能像 Server 版本一样通过连接主机

来进行远程控制。虽然最新版本的 VMware Workstation 集成了 VNC(Virtual Network Computing) Server 方便用户使用 VNC 客户端软件连接，但是后文中将论述该方法对于虚拟机的远程控制是不适用的。

总的来看，VMware 拥有完善的产品线，软件架构和特性根据定位对象不同而改变。针对本课题，VMware Server 提供了不错的解决方案，但是无奈该软件体积庞大占用资源过多，所以还需多加考虑。

2.2.2 VirtualBox

将视线转移到其他虚拟机产品，与 VMware 动辄几百兆大小的体积形成鲜明对比的是 VirtualBox 软件。VirtualBox 原由德国 Innotek 软件公司开发，该公司被 Sun 公司收购后将其命名为 Sun xVM VirtualBox。2010 年 1 月在 Oracle 收购 Sun 后，再次更名为 Oracle VM VirtualBox。与 VMware 最大的不同是，VirtualBox 是基于 GPL 的开源软件，虽然诞生较晚，但是更新速度很快，产品日渐趋于完善。

图 2.2 VirtualBox 的 GUI 界面(Version3.1.6)

与相同定位的 VMware 及 Virtual PC 相比，VirtualBox 的独特之处在于其支持 RDP(Remote Desktop Protocol)协议的远程访问、支持 iSCSI。另外，新版本的 VirtualBox 已经支持在客户机操作系统上的 USB 硬件装置。对 RDP 协议的支持正是本课题中对于虚拟机控制的基础，这在后文中将详细说明。

也许正是由于其开源软件的定位，VirtualBox 在设计时考虑到从事开源软件开发用户的需要，全面支持命令行的操作模式。VBoxManage[7]是 VirtualBox 的命令行接口。使用 VBoxManage，我们可以在主机操作系统的命令行中完全地控制 VirtualBox，这样极大地方便了开发人员在程序中对虚拟机的操控，如复制、开启、更改设置等。VBoxManage 支持 GUI 可访问的全部功能，甚至不限于此，VBoxManage 展示了虚拟化引擎的全部特征，包括 GUI 无法访问的。

选择 VirtualBox 作为本课题使用的虚拟机软件主要出于以下几方面的考虑：1)VirtualBox 是基于 GPL 的开源软件，开发人员对其使用不受限制；2)VirtualBox 支持 RDP 协议的远程访问，RDP 协议可以帮助在客户端在多种平台上实现对于虚拟机的访问；3)使用 VBoxHeadless 模式启动虚拟机，可以不显示 GUI 窗口而对虚拟机进行操控，这在多虚拟机情况下为服务器节省了很多资源；4)VirtualBox 的 VBoxManage 命令行接口可以帮助开发人员实现程序对虚拟机的操控。

2.3 虚拟机的配置

基于上述理由，在本课题中选择 VirtualBox 作为虚拟资源开发平台。下面要着手对 VirtualBox 进行配置，以符合课题开发的要求。配置包括与主机的文件共享和网络设置。

本课题的实验环境是主机为 Linux(内核版本 2.6.31)，在主机上安装虚拟机软件 VirtualBox(Version 3.1.6)，主机经 DHCP 分配 IP 地址连接到互联网。

2.3.1 主机文件共享

虚拟机的一个特性就是一台主机虚拟多个硬件系统来运行操作系统，这几个操作系统可以同时运行，互不干扰，形成有效的隔离。但同时带来一个问题，即虚拟机与主机逻辑上隔离导致文件共享不便，当很多个虚拟机需要同一个文件的时候需要复制到各自硬盘系统内，这就浪费了存储资源，与虚拟机的设计初衷相违背，并且在实际运行中各虚拟机之间的复制并不轻松。因此，配置虚拟机的首要任务就是要实现虚拟机与主机的文件共享。

VirtualBox 为上述情况提供了解决方案，通过安装增强功能来实现。VirtualBox 自带了一个增强工具 Sun VirtualBox Guest Additions，这是实现虚拟机与真实主机共享的关键。本文以 Windows XP 系统为例来说明这一过程。

启动虚拟 XP 后，点击控制菜单“设备”→“安装增强功能”，可以看到程序的安装界面，如图 2.3a 所示。然后设置主机中与虚拟机共享的文件夹。点击控制菜单“设备”→“分配数据空间”。进入对话框后先添加新的数据空间，设置“数据空间位置”时点击下拉列表，选择“其它”，这样才能在文件夹列表中找到主机中的文件夹，选择需要共享的文件夹后返回。勾选“固定分配”选项，现在我们在“数据空间”列表中就可以看到共享的主机文件夹了。

图2.3 a

图2.3 b

图2.3 c

图 2.3 XP 虚拟机与主机共享文件夹设置

上述工作完成后，VirtualBox 在主机上为虚拟机分配了一个数据空间，可以理解为为虚拟机增加了一个驱动器，下面的工作就相当于虚拟机对于该驱动器的加载。

在虚拟机中采用映射网络驱动器的形式来快速访问该数据空间。在虚拟机中打开“我的电脑”，进入后点击菜单“工具”→“映射网络驱动器”，进入后先指定驱动器号，如图 2.3b，接下来，点击浏览按钮，在“整个网络”树状列表中找到“VirtualBox Shared Folders”，该文件夹树下的地址即为“数据空间”中设置的主机共享文件夹。

完成以上设置后，虚拟机用户就可以快速访问主机的文件夹了，需要说明的是虚拟机对网络驱动器也有读写权限，所以 VirtualBox 通过这样的方式实现主机与虚拟机的文件共享，进一步方便服务器对多虚拟机的管理。

2.3.2 网络设置

VirtualBox 提供四种虚拟机接入网络的方式：NAT、Bridged Adapter、Internal、Host-only Adapter。下面具体说明这四种连接方式并选取最有利的。

1) NAT (Network Address Translation) 网络地址转换模式，这是最简单的实现虚拟机上网的方式。可以理解为虚拟机与网络的所有交互数据都由主机转发，虚拟机隐藏在主机背后，主机和网络中的任何机器都不能访问和查看到虚拟机的存在。
2) Bridged Adapter 桥接模式，虚拟机通过主机网卡假设一座桥连入网络。如果主机的网络环境是由 DHCP 分配 IP 地址，那么虚拟机也可以分配到真实网络中的独立 IP，这样虚拟机就与网络中的真实机器无异。
3) Internal 内网模式，这种模式下虚拟机与外网完全断开，只可以实现虚拟机与虚拟机之间的内部网络模式。
4) Host-only Adapter 主机模式，这是一种比较复杂的模式，通过对虚拟机和网卡不同的设置来实现不同的功能，实际上以上三种连接模式通过主机模式都可以实现。

四种连接方式下主机、虚拟机和网络之间的关系表 2.1 所示，NAT 模式虚拟机获得虚拟 IP，所以外部网络(包括主机)无法访问该虚拟机；桥接模式使得虚拟机获得独立真实 IP，所以与网络内的计算机(包括虚拟机)都可以互相访问；内网模式下虚拟机与外网完全断开，如果将两台虚拟机设为同一网络则可以互相访

问；主机模式下情况较为复杂，虚拟机可以被设置在不同于主机的网络来连入互联网，下表所列为默认情况下访问关系。

表 2.1 VirtualBox 四种网络设置虚拟机与主机、网络的访问关系

	NAT	Bridged Adapter	Internal	Host-only Adapter
虚拟机访问主机	√	√	×	×
主机访问虚拟机	×	√	×	×
虚拟机访问网络其他主机	√	√	×	×
网络其他主机访问虚拟机	×	√	×	×
虚拟机访问虚拟机	×	√	√	√
IP	10.0.2.15	DHCP 分配	192.168.56.*	--
网关	10.0.2.2	与主机相同	--	--
DNS	10.0.2.3	与主机相同	--	--

用户可以针对各自应用选用不同的网络连接方式，NAT 配置简单，且对外网不可见，适合 IP 资源紧张的用户；桥接模式对外网开放，适合作为服务器使用；内网模式适合在虚拟机内部搭建网络；主机模式适合搭建更为复杂的网络环境。上文中提到 VirtualBox 支持 RDP 远程访问，实际是将虚拟机作为主机的一个端口开放给外网，因此不管选择哪种网络连接方式，外网计算机(包括主机)都可以通过 RDP 远程控制虚拟机，当然远程控制的意义不同于享受虚拟机提供的网络服务。针对本课题，考虑到虚拟资源的共享意义不在于虚拟机提供 web 服务而在于对虚拟机本身的操控，所以选择简单易用且不占用 IP 资源的 NAT 连接模式。

选定网络连接模式后，大多数情况下接受默认的设定就能使虚拟机良好地工作。需要注意到的是，VirtualBox 为每个虚拟机提供最多 8 张虚拟网卡的支持，GUI 界面下可以详细配置前四张，命令行下的 VBoxManage 能够配置全部 8 张网卡，详见用户手册[8]，此处不再赘述。

在完成上述设置后，VirtualBox 提供的虚拟机就能很好地为用户服务，用户可以将虚拟机打造成为服务器，或是适合计算、编译的主机等，这样可以在虚拟

社区内形成良好的共享机制，促进提高个人计算机的利用率和整个网络的计算、存储能力。

2.4 小结

本章首先概述了虚拟机的发展、现状和主要作用。然后比较目前主流的虚拟机软件，针对本课题的技术特点选取 VirtualBox 作为虚拟机支持。

VirtualBox 支持 RDP 协议远程访问，其 VBoxHeadless 模式便于无图形界面的管理，VBoxManage 命令行特性适合程序开发的需求，另外，VirtualBox 是基于 GPL 的开源软件。

在虚拟机创建之后需要对其进行相应设置，以满足本课题的要求，主要包括共享文件夹的设置和网络配置。

本章为后续的研究过程打下了基础，文中 RDP 协议、VBoxManage 等特性将在第 3 章、第 4 章做出具体说明。

第3章 基于 web 浏览器访问虚拟机

第 2 章中针对本课题的需求选择了虚拟机并进行配置，本章将详细说明基于 web 浏览器访问虚拟机的实现方法。首先介绍尝试的两种方案：基于 VNC 的远程访问和基于 RDP 的远程访问，然后针对这两种方案的优缺点做出取舍并加以完善，以此来实现用户对虚拟机的便捷访问。

3.1 功能描述

在云计算中，硬件资源和软件资源经过虚拟化由“云”这一超级计算机所掌控，用户在获得访问权限后可以在网络中“找到”这些资源。虚拟资源并不是虚无缥缈杳无踪迹的，落到实处仍然是以计算机为介质存在，只不过这计算机可能是虚拟出来的。在虚拟社区中也是如此，用户获准对某资源(如软件、存储空间)的使用权限后必须要访问并操控该虚拟机。

由于用户端的平台多种多样，可以是 Windows、Linux、Mac 各个操作系统，可以是个人计算机或者智能手机，所以远程访问的客户端对多平台的普适性显得尤为重要。Web 浏览器无疑提供了一个很好的解决方案，它支持的 HTTP 协议和 HTML 语言等在网络世界里是通用的，其越来越强大的功能也使得 web 浏览器的使用变得简单便捷。作为 web 应用程序的助手，Java 平台的通用性也保证了基于 web 浏览器远程访问的通用性和可靠性。本章想要实现的功能是用户通过对浏览器指定被访问虚拟机所在主机的 IP 和相应端口来实现对虚拟机的操控。

3.2 基于 VNC 的远程访问

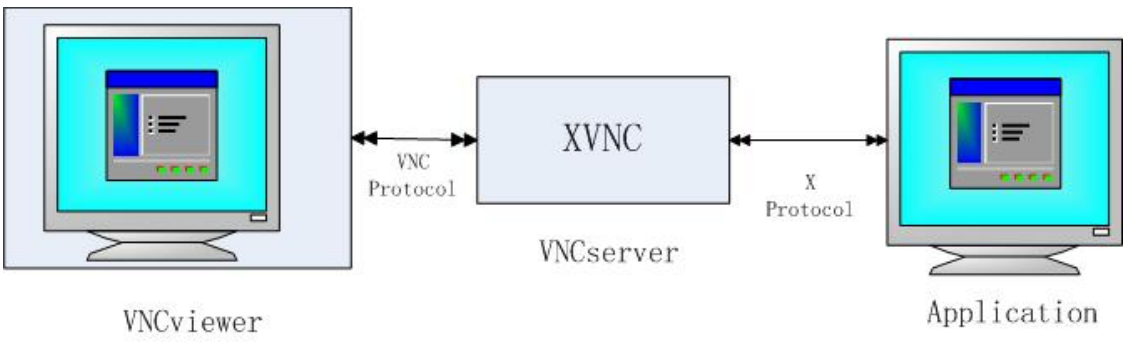

图 3.1 VNC 的工作原理

VNC(Virtual Network Computing 虚拟网络计算)是由 AT&T 的欧洲研究实验室开发的远程控制软件，可以将完整的窗口界面通过网络传送到另一台计算机的屏幕上。VNC 的客户端和服务端可以是不同类型的平台，这保证了各操作系统之间的通用性。VNC 实现的核心要素是用户使用 VNC 客户端连接到运行 VNC 服务器机器上时，客户机通过键盘和鼠标动作来执行存放在服务器上的应用程序，服务器桌面的快照被压缩并且通过其通信协议 RFB(Remote Frame Buffers)协议发送到客户端[9]。其工作原理如图 3.1 所示。

目前 VNC 有两个发行版本，基于 Linux 和 Unix 操作系统的免费开源版和付费的 RealVNC。后者 VNC Enterprise Edition for Windows 是工业标准 VNC 的增强版，，提供了核心的 Enterprise Edition 安全增强，包括 2048 位 RSA 服务器验证和 128 位 AES 会话加密术，可以使用证书验证保证安全。后者还有一些为 Windows 网络量身定做的许多其他的特性，如集成了 Windows 防火墙和文件传输功能等。Linux 操作系统下没有这样的增强版本，所以其安全性让人担忧。当 VNC 建立连接后，在客户端和服务器之间传输的数据是没有加密的，可能会被监听或嗅探。目前较为普遍的解决方案是为 VNC 协议开辟一条专用隧道，如使用 SSH 为数据加密。或者要实现更高的安全性能，可以考虑用基于 SSL 的会话密钥加密机制[9]。

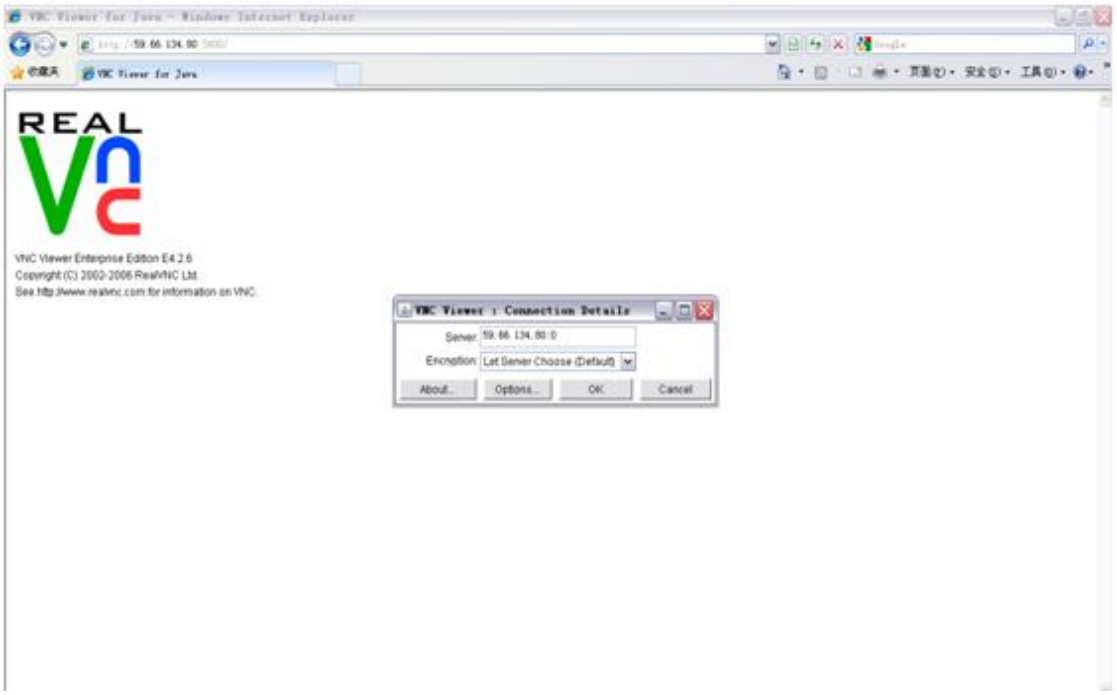

图 3.2 VNC 使用浏览器进行远程访问

VNC 远程访问软件的过人之处在于其支持浏览器的访问。VNC server 在启动时加载了 web 控制端，采用 5800 端口 http 协议。客户端不需安装 VNC viewer 就可以通过浏览器访问被控端。但需要说明的是，浏览器需要 Java 虚拟机的支持才能实现这个功能。VNC 在服务器与浏览器之间使用的是 TCP/IP 连接。当浏览器与服务器建立连接后，实际上浏览器调用了 Java 虚拟机完成后续的显示并操作等应用。图 3.2 显示了 VNC 使用浏览器进行远程访问的开始画面，在通过服务器端验证后就弹出 java 虚拟机的应用框，在应用框内完成对被控端的访问。

虽然 VNC 是一款成熟的远程访问软件，支持多平台之间的互通，但是在考察了其访问特点后发现并不适合本课题的使用。首先需要在每个虚拟机里都安装并配置好 VNC server，这增加了服务器的工作量。其次 VNC 访问需要每个虚拟机具有真实的 IP，或者在 DNS 服务器中注册域名，这就需要虚拟机采用桥接的方式，对于 IP 资源紧张的个人用户来说并不可取。

3.3 基于 RDP 的远程访问

在第 2 章选择虚拟机软件的时候已经提到 VirtualBox 支持 RDP 协议的远程访问，这对于本课题提供了另一种解决思路。

3.3.1 VRDP 简述

VirtualBox 内置了支持 VRDP(VirtualBox Remote Desktop Protocol)的服务器。VRDP 允许用户在本地计算机看到远程计算机的每个虚拟机的窗口输出并实现操控，就好像是在本地运行虚拟机[10]。

VRDP 是对微软的 RDP 协议的向后兼容扩展。该协议将显卡和声卡的更新从远程计算机发送到本地，将本地的鼠标和键盘事件发送回去。所以用户可以用标准的 RDP 客户端来远程控制虚拟机。

3.3.2 Rdesktop 对于虚拟机的访问

Rdesktop[11]是面向 RDP 协议的开源客户端，设计初衷是为了向 Linux/Unix 系统提供访问 Windows 桌面的能力。目前大多数 Linux 系统已经集成了 Rdesktop。

在 VirtualBox 中，VRDP 服务器使用标准的 RDP TCP 端口 3389.，如果 Windows 系统的 RDP 服务器已经占用这个端口，用户需要作出修改。同理，一台主机上安装多台虚拟机时，每个虚拟机需要对应不同的端口，5000 到 5050 端口通常处于闲置状态，建议使用这些端口。

VirtualBox 推出无头模式(VBoxHeadless)来配合 VRDP 的使用。当服务器启动很多虚拟机时，如果全都打开图形界面将会占用大量的系统资源，或者当图形界面不是必须的情况下，可以采用 VBoxHeadless 模式来启动虚拟机。如果本地服务器想要控制该虚拟机，可以采用 rdesktop localhost : port 的形式来实现。

当然，服务器开启多台虚拟机的目的是为了方便客户端的使用，或者在虚拟社区内共享资源时方便其他用户使用，Rdesktop 使得这一过程变得简单。客户端在获得对某服务器的某一台虚拟机的使用权限后，根据服务器域名(或者 IP)加上该虚拟机的端口就可以实现连接。当然，基于安全性的考虑，服务器不会开放 IP 或者端口，安全的解决方案将在第 4 章讲述。

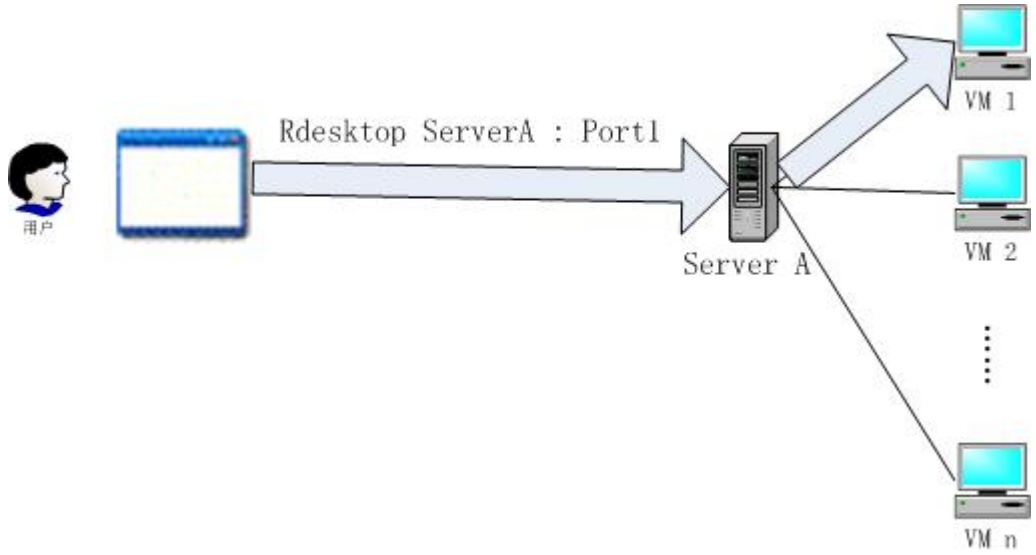

图 3.3 使用 Rdesktop 访问远程虚拟机

这种方案的好处有三个方面：1) Rdesktop 方便易用，VirtualBox 本身支持 RDP 协议；2) 服务器的每台虚拟机可以采用灵活的联网方式，不受制于必须拥有真实 IP 才能被访问的限制；3) 服务器容易控制对每台虚拟机的访问权限，只需管理相应的端口即可。

3.4 Web 浏览器访问虚拟机

在确定了使用 RDP 协议访问虚拟机后，需要着手对 Rdesktop 进行改进以满足浏览器的需求。可以使用 Java Applet 帮助浏览器加载应用程序。在此之前，浏览器需要安装 Java 插件以获得 Java 虚拟机的支持。

3.4.1 功能描述

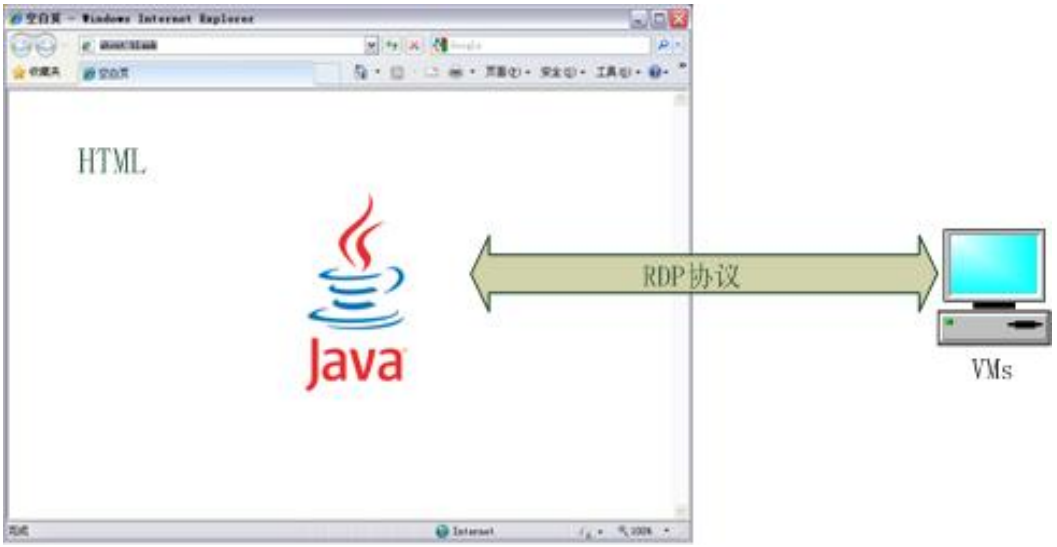

图 3.4 浏览器调用 Java Applet 访问虚拟机

当用户浏览器打开访问虚拟机的 HTML 页面时，HTML 加载嵌入在内的 applet，applet 调用 Java 虚拟应用程序使用 RDP 协议连接目标虚拟机。整个过程需要浏览器、Java 虚拟机和 RDP 访问协议的配合，其关系如图 3.4 所示。

用户需要向 HTML 页面提交被访问服务器的域名(或者 IP)和被访问虚拟机在主机的端口，HTML 将该数据转到 Applet 中，之后的过程就与 Rdesktop 的远程访问过程无异。

3.4.2 功能实现

Rdesktop 是开放源代码的软件，所以我们能在其基础上作出修改以增强其功能。Rdesktop 系列包含 java 版本，这方便了开发人员对其扩展性应用。但是真正实现 Applet 功能，还需要这个 java 文件进行包装，并启用 applet 线程来调用。图 3.5 显示了具体实现的结构图。

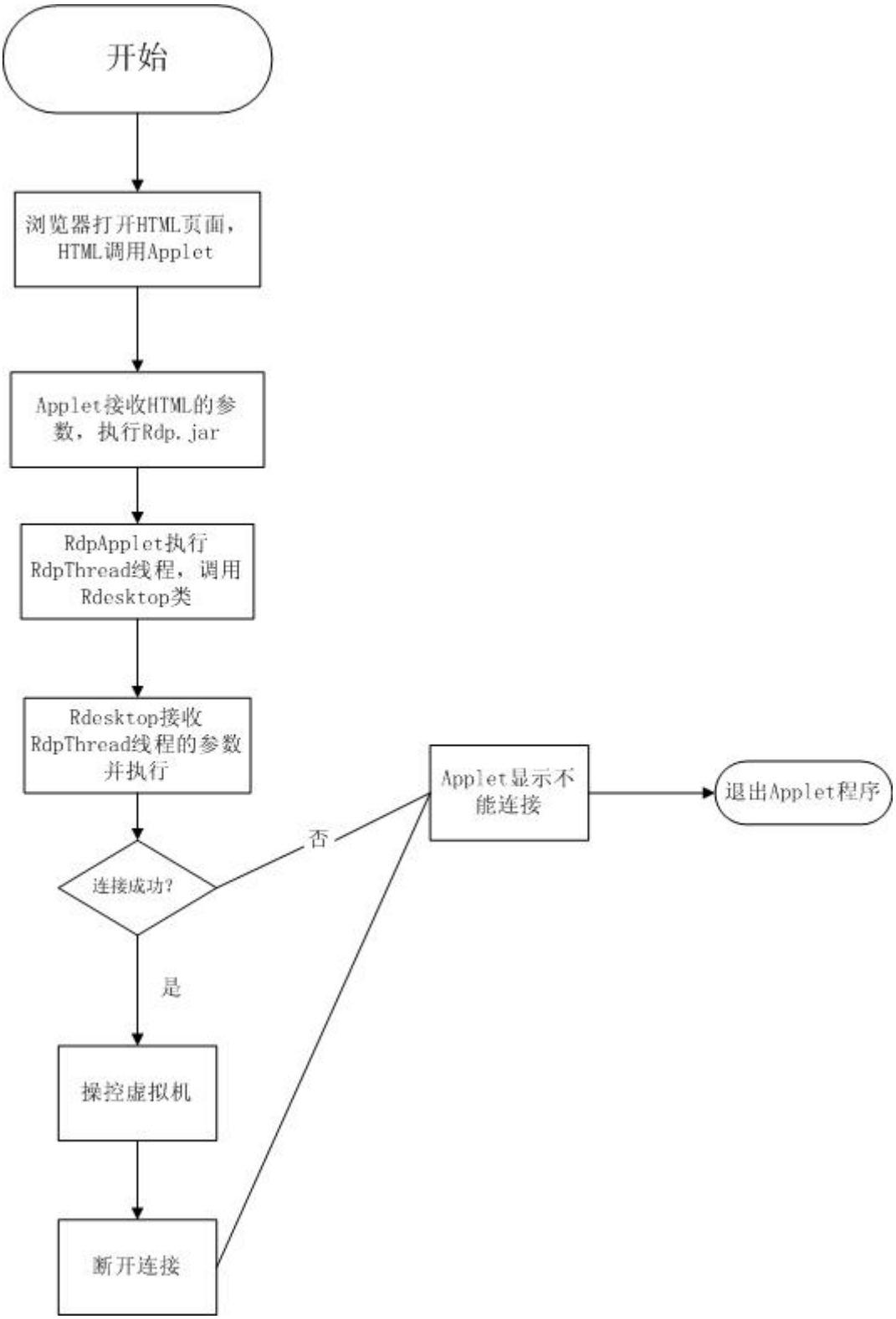

图 3.5 浏览器调用 Java Applet 实现远程访问流程

Applet 应用程序必须嵌入在 HTML 页面中，才能被解释执行，同时 Applet 需要从 web 页面获得参数，所以还需要编写 HTML 文件代码。与 Java 等编程语言不同的是，HTML 文件在编写好之后可以直接运行不需要编译。HTML 文件中至少需要以下信息：

1) 必须包含一段<applet>和</applet>标记的代码，这两个标记之间的代码标识 Applet 运行的一系列参数，针对本课题的程序需要标注被访问服务器的域名(或者 IP)、被访问虚拟机的端口和窗口分辨率大小等；
2) 编译生成的 class 文件名及调用路径，如果是在 jar 包里需要指明 jar 内路径，这个 class 文件就是 applet 被解释后调用的程序文件。

使用 Java Applet 需要注意 applet 的权限，因为 applet 通常被设计为从远程站点下载然后再本地执行，当用户在浏览器中启动 java，就会立即执行 applet 代码，用户不能确认或者停止 applet 的运行。针对这种情况，java 虚拟机将 applet 的执行环境限制在“沙箱(sandbox)”内，运行在“沙箱”中的 applet 不能更改或探查用户的系统。但是在本课题中，applet 需要使用 TCP 进行通信，如果不加措施的执行，就会触犯安全规则被当作 SecurityException 异常而阻止。解决这一问题的途径是对 applet 进行签名，被签名的 applet 带有一个能表明签名者特征的证书。下面简单说明签名过程。

```
seanvb@seanvb-laptop:~/java$ keytool -genkey -alias seanj -keystore seanjava -va
lidity 2000
输入keystore密码：
再次输入新密码：
您的名字与姓氏是什么？
  [Unknown]：  Sean
您的组织单位名称是什么？
  [Unknown]：  Tsinghua
您的组织名称是什么？
  [Unknown]：  Tsinghua
您所在的城市或区域名称是什么？
  [Unknown]：  Beijing
您所在的州或省份名称是什么？
  [Unknown]：  Beijing
该单位的两字母国家代码是什么
  [Unknown]：  CN
CN=Sean, OU=Tsinghua, O=Tsinghua, L=Beijing, ST=Beijing, C=CN 正确吗？
  [否]：  是

输入<seanj>的主密码
        （如果和 keystore 密码相同，按回车）：
```

图 3.6 使用 keytool 创建密钥

由于所有的 class 文件都被封装在 rdptmp.jar 文件中所以需要对 jar 文件签名，使用 java 签名工具 jarsigner。jarsigner 使用来自密钥仓库的密钥和证书信息为 jar 文件生成数字签名。密钥仓库是一个由私钥及其相关的 X.509 证书链（它鉴别相应公钥）组成的数据库。使用 keytool 实用程序来创建和管理密钥仓库，jarsigner 使用实体的私钥创建签名。已签名的 jar 文件包含一份来自密钥仓库的公钥（它对应于用于为该文件签名的私钥）的证书副本。jarsigner 可以使用已签名的 jar 文件中的证书（在其签名块文件中）来校验其数字签名[12]。

以 desktop.jar 文件为例，首先使用 jarsigner -verify -verbose desktop.jar 命令来校验该文件的签名信息，对于一个未签名的文件来说得到的结果是“jar 未签名。（缺少签名或签名无法解析）”。接下来使用 keytool 命令来创建密钥：keytool -genkey -alias seanj -keystore seanjava -validity 2000。命令中 seanj 和 seanjava 分别为密钥名字和别名，2000 为有效期，单位是天。得到结果如图 3.6 所示。

最后是使用这个密钥对 desktop.jar 文件签名：jarsigner -keystore seanjava desktop.jar seanj。注意该命令的格式，seanjava 和 seanj 两个名字分别要和创建密钥时的相对应。

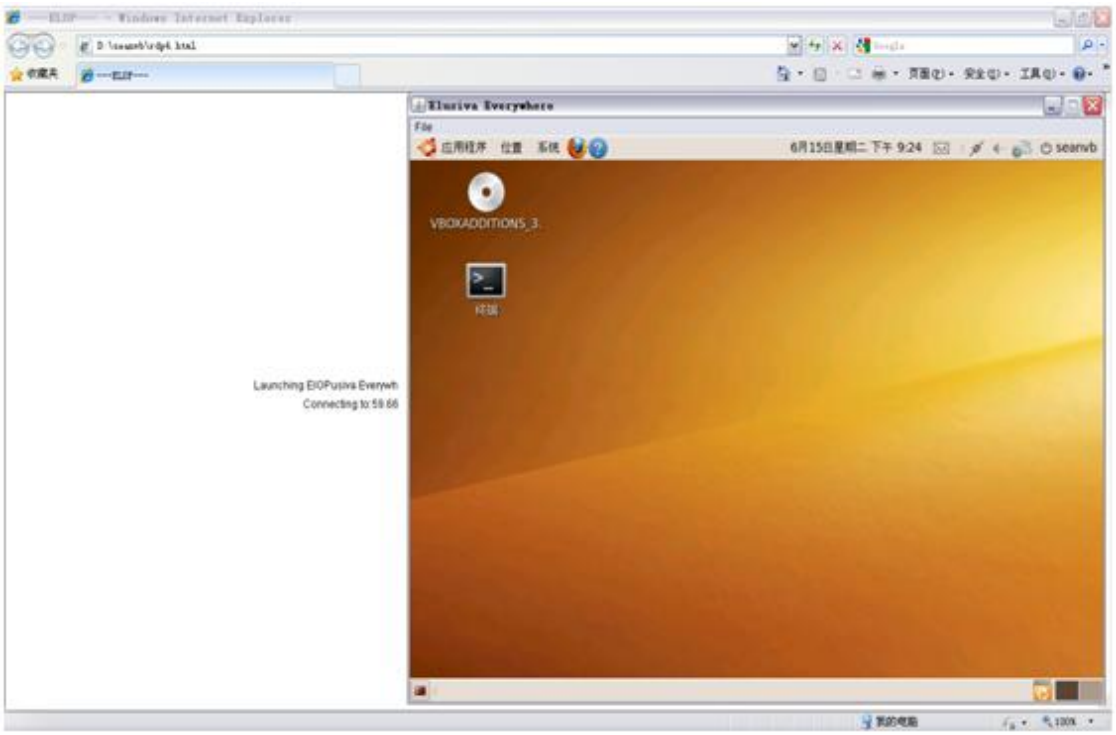

图 3.7 浏览器使用 applet 访问远程虚拟机

在完成签名后，用户就可以在执行 applet 前校验签名者的信息，并选择是否接受 applet 的执行。如果用户选择信任该签名，就可以运行 RDP 协议对远程虚拟机进行访问。结果如图 3.7 所示。

3.5 小结

本章介绍了基于 web 浏览器访问虚拟机的方法。首先描述了虚拟环境下对远程访问虚拟机的特点，在该特点的基础上提出了课题要实现的功能。基于 web 浏览器的访问可以保证客户端在各平台的通用性。

随后介绍了基于 VNC 的远程访问。VNC 可以实现多平台计算机的互相连通，而且支持浏览器远程访问，在做出基于 SSH 或者 SSL 的功能增强后可以保证连通数据的安全。但是使用 VNC 远程访问需要在虚拟机内安装并配置 VNC Server，并且要求每台虚拟机具有真实的 IP 地址，这不适合本课题中虚拟机的网络架构。

我们可以利用的是 VirtualBox 支持的 VRDP 服务，使用 Rdesktop 可以方便的访问一台服务器上的任意虚拟机，并且由于 Java 平台的通用性，基于 web 浏览器的 RDP 远程访问能满足不同操作系统的需求。RDP 协议访问是基于被访问主机的域名(或者 IP)加上被访问虚拟机占用的端口，所以虚拟机的网络可以灵活配置不受影响。

本章最后介绍了 RDP 协议在 Java Applet 的实现，给出了浏览器访问远程访问虚拟机的结果。但是这种连接是不安全的，在第 4 章将会讲述虚拟机访问的权限控制和安全连接。

第4章 虚拟机的安全共享

第 3 章完成了基于 web 浏览器的远程访问，但是这种访问方式太过简洁以至于非常不安全。任何人在获取服务器的 IP 地址和虚拟机占用的端口后都能对其进行访问，虽然 VirtualBox 提供了验证过程，但只是输入主机的用户名和密码，这很容易被绕过，而且连接后所有的数据都采用明文传送，极易被窃取。

本章将针对远程虚拟机的安全问题提出解决方案，重点在于访问权限的控制和建立安全连接。

4.1 系统结构

在虚拟社区内，用户根据兴趣或者工作需求形成许多虚拟组织 VO (Virtual Organization)，在每个虚拟组织内用户可以采取某种信任机制将自己的虚拟机授权给其他成员使用，这就是虚拟社区内的共享概念。

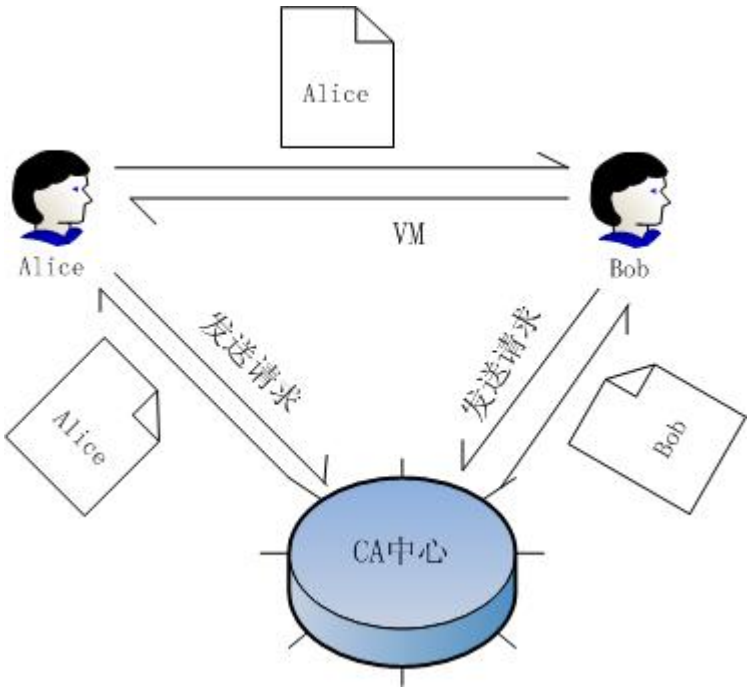

图 4.1 虚拟社区内虚拟机共享示意

如图 4.1 所示，Alice 和 Bob 都加入了某虚拟组织，认证中心(CA，certification authority)向他们分别颁发了证书，证书由 CA 签名，提供了 Alice 的身份与 CA 身份之间的绑定。在 Bob 授权 Alice 使用自己的虚拟机之后，Alice 使用带有个人身份的证书去申请与 Bob 的连接，连接地址由服务器派发给 Alice。Bob 再使用 CA 公钥验证 Alice 传送来的证书以确认 Alice 的身份，随后对 Alice 的公钥建立信任。在 Bob 授予虚拟机使用权限时，需要确定 Alice 可以使用的虚拟机的端口号，这样在建立连接后向其开放该端口，但是其他端口依然对其封闭。这需要防火墙的配合。通过上述过程，在虚拟社区内 Bob 将自己的虚拟机安全地共享给 Alice 使用。

4.1.1 数字证书和认证

实现访问权限控制的基础是数字证书的认证，证书用来确保连接对象是可信的，这就需要一个包括第三方审查和提供用户身份的可信系统。公钥基础结构(PKI，public-key infrastructure)可以建立和支持这个可信系统。

PKI 允许用户相互验证各自使用的、由认证中心颁发的数字证书。下面举例说明本课题中使用的 CA 颁发证书过程。

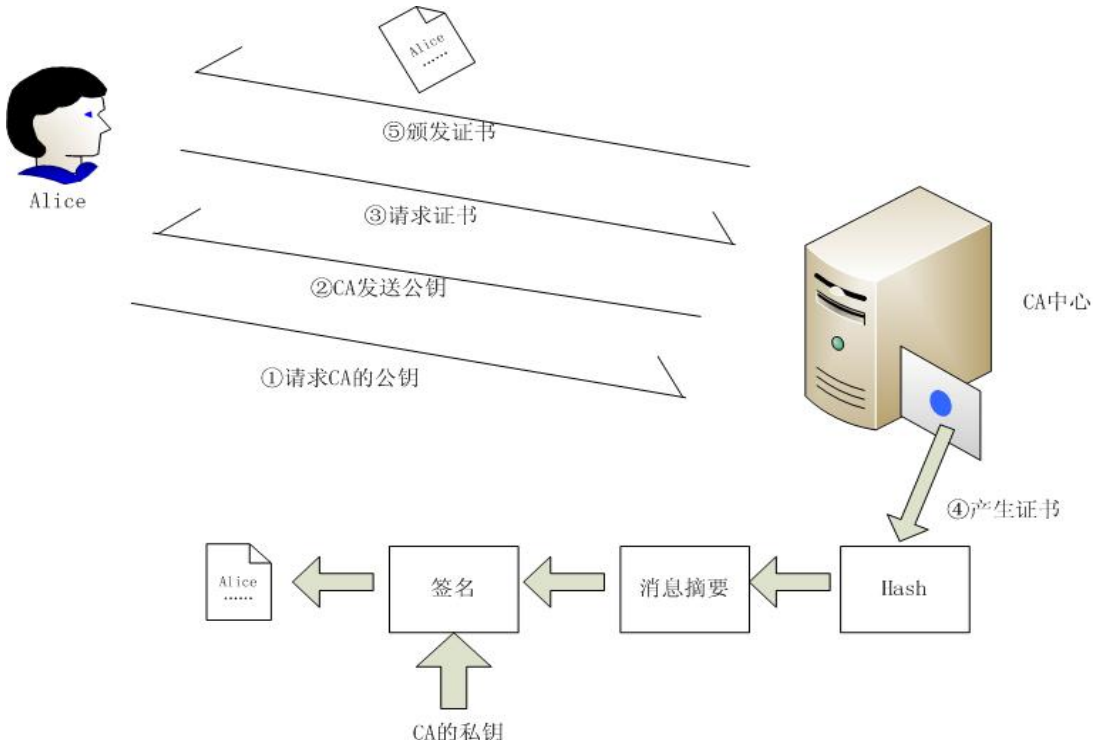

图 4.2 CA 中心颁发证书

① Alice 向证书服务器提出申请，请求 CA 的公钥；

② CA 服务器将自己的根证书回复给 Alice；

③ Alice 提出证书请求，在请求中包含 Alice 的身份信息和公钥，该请求需要 CA 公钥签名；

④ CA 服务器接收证书请求，首先验证 Alice 的身份信息，然后将其身份和公钥绑定，产生一个数字证书，该证书由 CA 服务器签名；

⑤ CA 服务器向 Alice 颁发证书。

Alice 获得 CA 服务器签发的证书后，就可以传送给其他用户以获取信任。其他用户使用 CA 公钥验证 Alice 的证书，在人为授予权限后，就可以对 Alice 的公钥建立信任连接。

4.1.2 OpenVPN 建立安全连接

VPN(Virtual Private Network)，即虚拟专用网络。顾名思义，这个专用网络是虚拟的，但能达到网络专用的功能，它可以在公共网络中将数据经由“加密管道”传送，从而达到私有专用的目的。

VPN 诞生的原因就在于人们对于网络中传输数据安全性的关注，目前组建 VPN 虚拟专用网主要依靠四项技术来保证安全：隧道技术、加解密技术、密钥管理技术和身份认证技术[13]。

OpenVPN 就是一款用于创建虚拟专用网络加密通道的软件包，它允许使用证书进行身份验证。实际上，OpenVPN 的数据加密是基于 OpenSSL 的，即 OpenVPN 使用 OpenSSL 库进行加密和验证，所以 OpenSSL 支持的加密算法它都能够使用。我们知道，SSL 协议在应用层协议通信之前就已经完成加密算法、通信密钥的协商和服务器认证的工作。在此之后应用层传送的数据都将被加密，从而保证数据的私密性。承接上一小节的内容，在本课题中，我们将用到 OpenVPN 对证书的验证功能。

4.2 功能实现

本节将探讨通过布署防火墙和 OpenVPN 以实现虚拟机访问的权限控制和安全连接。整个系统分为客户端和服务器端，效果如图 4.3 所示。

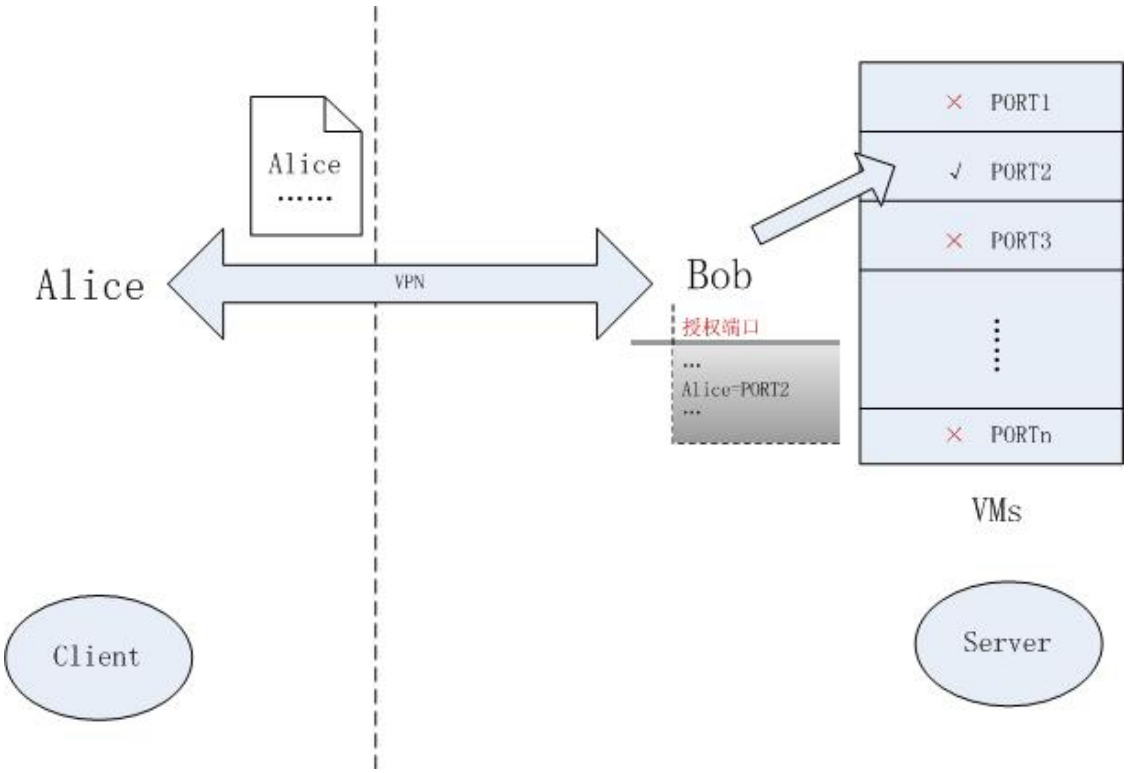

图 4.3 共享实现

4.2.1 服务器端

服务器指被共享虚拟机所在的主机，在云计算中可以是提供虚拟资源的服务器集群，在虚拟社区内每位提供虚拟机共享的成员都可以作为服务器。由于 VirtualBox 的访问模式是主机的域名(或 IP)加上被访问虚拟机占用的主机端口，虚拟机访问权限的控制就可以简化为对服务器端口访问的控制。

实现端口访问控制的前提是开启防火墙。在 Linux 系统中 iptables 提供了强大的防火墙功能，这是一个 IP 信息包过滤系统，可用于添加、编辑和删除防火墙对于信息包过滤所遵循的规则。iptables 经过适当配置后，就可以可靠地保护 Linux 系统。但是 iptables 的配置对于普通用户来说显得复杂，所以推荐使用 iptables 的配置工具——FireHol。FireHol 自称是“表达防火墙规则的语言”，的确，使用 FireHol 几个简单的命令就可以完成基于防火墙的 iptables 配置。

在 FireHol 安装完毕后使用 firehol helpme > out 生成基本配置文件，根据本课题的要求需要在 wan 网增加 https 的连接，另外根据实际需求开启 ssh、ICMP 等服务。

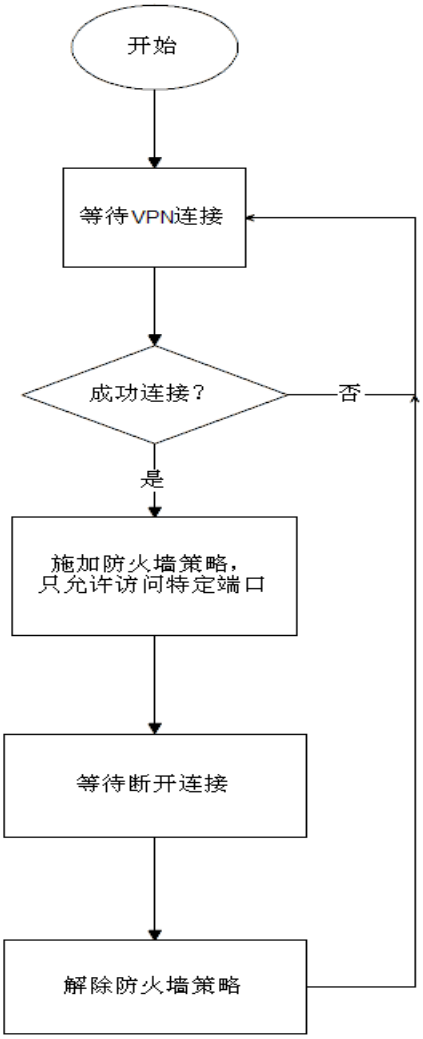

图 4.4 服务器端流程图

FireHol 配置完毕后，使用 firehol start 的命令开启防火墙，注意这需要 root 权限。防火墙开启后，所有虚拟机所占用的端口都被保护，不能被访问。如何开启某端口的访问权限，这需要 OpenVPN 的配合。

首先在服务器上安装 OpenVPN，这是基于 GPL 的开源软件，在 Linux 下的安装非常方便，此处不再赘述。安装完毕后，首先需要增加服务端用户，然后进行配置。编辑/etc/openvpn/openvpn.conf 文件，此处将端口设为 443，TCP 协议。443 端口提供 https 服务，其安全基础即 SSL，在 TCP 连接之前需要身份验证并对连接加密。使用 OpenVPN 进行身份验证需要 CA 的公钥和 CA 颁发给用户的证书，在配置文件中设置公钥证书的访问路径。另外，如果服务器本身已经生成了 OpenSSL 的 key，那么在配置文件中指明路径，否则使用 openssl dhparam -out 命令生成 key 文件并指明其路径。

OpenVPN 配置完毕后开启 OpenVPN 服务，客户端就可以与服务器端建立安全连接。在配置文件中设定可以同时连接的客户端个数，此处设定为 30，允许 30 个客户端同时使用 OpenVPN 连接服务器建立私有专用网络。

需要注意的是，并不是所有的证书验证通过就能获得端口访问权限，证书验证通过只能说明对方与自己来自同一个组织，这里需要增加用户访问端口的权限。读取证书后获得证书 CN(common name)，使用 CN=PORT 的格式添加使用权限，如授权成员 Alice 端口 3399 的访问权限，就可以在/etc/openvpn/clients.rc 添加 Alice=3399。Clients.rc 文件作为程序的接口，方便上层程序添加，编辑授权信息。

4.2.2 客户端

由于需要与服务器建立虚拟专用网络连接，所以客户端也需要安装 OpenVPN 软件，安装完毕后需要指明服务器的域名和建立远程的端口，在本课题中，服务器(即共享虚拟机的主机)的域名(或者 IP)由虚拟社区的管理服务器提供，端口与服务器保持一致采用 443。

客户端在申请连接时需要加载自己的证书，使用明文传送的方式传递到服务器端，等待服务器验证身份。身份验证通过后，客户端就与服务器端建立了一条“专用网络”，所有数据都经过加密传送，这保证了数据传送的隐私性和完整性。具体到本课题中，在这条专用网络中，服务器的 IP 地址为 10.8.0.1，客户端使用这个 IP 地址进行连接，使用真实的 IP 连接则会被拒绝。

由于服务器端只开启了有限的几个端口，其他端口被保护不能连接，所以客户端将证书传递到 443 端口后，服务器根据本地权限控制策略为该用户开启相应端口，由于 OpenVPN 建立了与客户端的专用网络，所以该端口只对授权用户开放，其他用户依然不能访问。

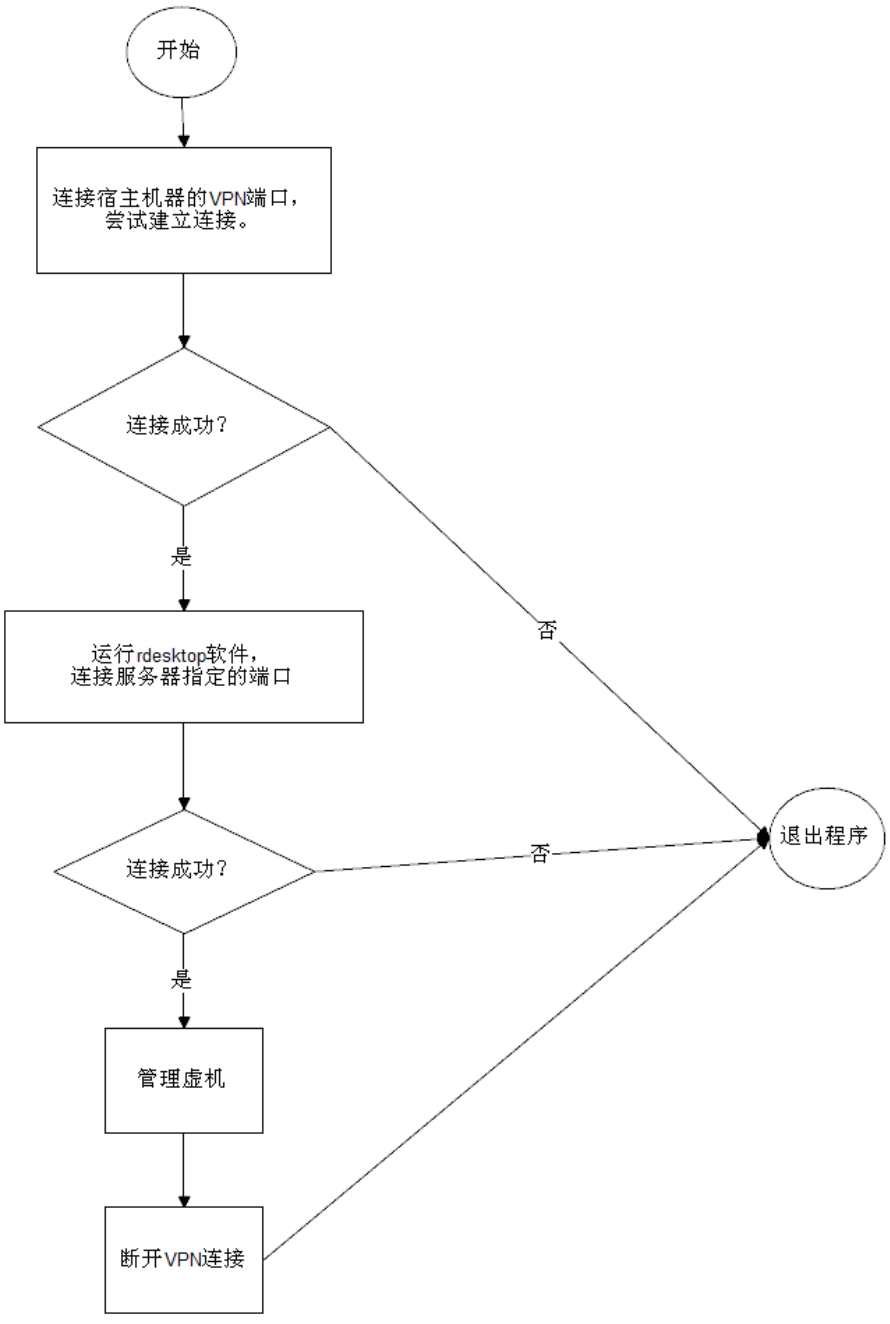

图 4.5 客户端程序流程

4.2.3 结果展示

在课题实验中，服务器端开启了两台虚拟机，操作系统分别为 Windows XP 和 Ubuntu，占用主机端口分别为 3389 和 3391。然后开启防火墙和 OpenVPN 服务，等待客户端的连接。

客户端使用 CA 服务器颁发的证书去连接服务器，OpenVPN 判断证书验证过程，有的证书需要私钥密码才能打开。服务器验证身份后与客户端建立虚拟专用网络通道，如果服务器的端口访问控制策略里授权该用户使用某端口，则可以访问该端口，即可以远程控制该端口的虚拟机。

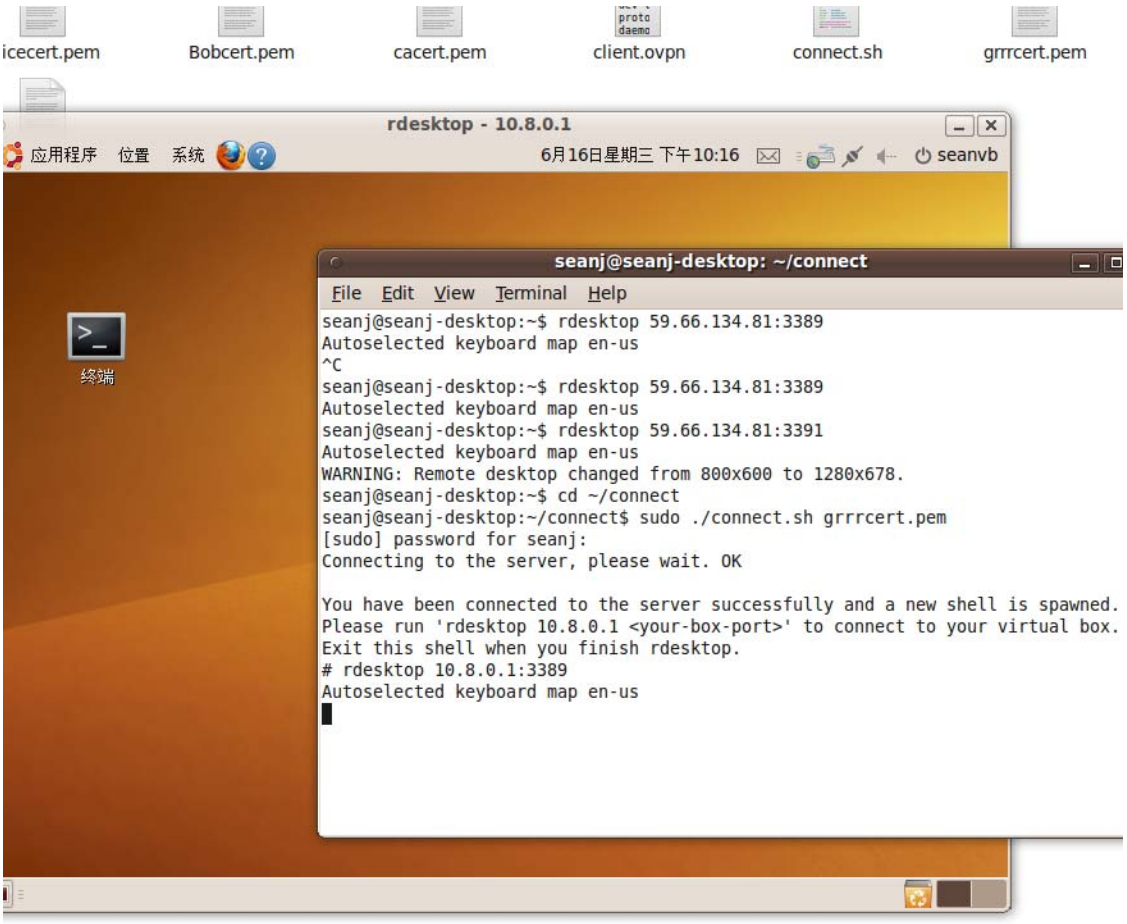

图 4.6 使用被授权的证书访问虚拟机

如图 4.6 所示，使用证书 grrrcert.pem 连接服务器，身份验证后建立了安全连接，并开启了一个 shell，只要 shell 未关闭就可以使用任意 RDP 工具访问开启的

端口，当然也包括第 3 章的 web 浏览器。本例中服务器为证书 grrrcert.pem 授权了端口 3389 和 3391，所以可以使用 rdesktop 连接 10.8.0.1 的 3389 端口。客户端与服务器主机之间建立的安全通道，服务器端被定义为 10.8.0.1 这一虚拟 IP，使用服务器真实 IP 连接则被防火墙拒绝。

服务器同时允许接纳最多 30 个连接，与每一个客户端的连接都是独立的，这就意味着为用户 grrrcert.pem 开启的端口不会对未授予权限的用户开放。如图 4.7 所示，服务器为 alicecert.pem 授予 3389 端口的使用权限，alicecert.pem 可以正常访问该端口的虚拟机，但是对于未授权的 3391 端口则会被服务器拒绝，尽管此时 3391 端口已经为用户 grrrcert.pem 开放。

```
# exit
Disconnecting...  Done
seanj@seanj-desktop:~/connect$ sudo ./connect.sh alicecert.pem
Connecting to the server, please waitEnter Private Key Password:
. OK

You have been connected to the server successfully and a new shell is spawned.
Please run 'rdesktop 10.8.0.1 <your-box-port>' to connect to your virtual box.
Exit this shell when you finish rdesktop.
# rdesktop 10.8.0.1:3389
Autoselected keyboard map en-us
# rdesktop 10.8.0.1:3391
Autoselected keyboard map en-us
ERROR: 10.8.0.1: unable to connect
#
```

图 4.7 使用证书访问未授权的虚拟机

在图 4.8 中，服务器并无 tomcert.pem 的信息，尽管该证书由同一 CA 服务器颁发，也可以通过身份验证，但是所有的连接请求都被过滤。

```
seanj@seanj-desktop:~/connect$ sudo ./connect.sh tomcert.pem
Connecting to the server, please waitEnter Private Key Password:
. OK

You have been connected to the server successfully and a new shell is spawned.
Please run 'rdesktop 10.8.0.1 <your-box-port>' to connect to your virtual box.
Exit this shell when you finish rdesktop.
# rdesktop 10.8.0.1:3389
Autoselected keyboard map en-us
^C
# rdesktop 10.8.0.1:3391
Autoselected keyboard map en-us
^C
#
```

图 4.8 使用未授权证书访问

至此，使用防火墙和 OpenVPN 使虚拟机的共享安全和可靠，即控制了访问权限，又保证了数据传送的安全。

4.3 小结

本章讨论了虚拟机的安全共享的方法，采用数字证书的认证方式验证身份，根据服务器端口控制策略允许授权用户接入相应端口，实现权限控制功能；使用 OpenVPN 建立安全连接通道，保证数据传送的安全性。

本章随后具体介绍了在服务器端部署防火墙和 OpenVPN 并编写脚本实现服务器的自动控制，然后介绍了在客户端使用证书连接的方法。

最后的实验结果展示了采用这种方法可以很好地控制访问权限和建立安全连接，基本保证了虚拟机共享的安全。

第5章 全文总结及展望

5.1 课题成果总结

本课题研究中，我主要完成了以下几个方面的工作：

1) 针对课题背景考察并选取了适合的虚拟机软件，并针对课题特点完成了相应的配置；

2) 选取 RDP 作为远程访问虚拟机的工具，使用 Java Applet 为浏览器加载应用程序以便用户使用 web 浏览器访问远程虚拟机，方便了跨平台用户使用虚拟资源。

3) 设计虚拟机安全共享的系统结构，使用数字证书认证的方式通过 OpenVPN 建立安全连接，分别在客户端和服务器端实现了连接和部署，保证了虚拟机共享的安全。

总的来说，我较好的完成了本次毕业设计的预定目标，并为后续相关工作打下了良好的基础。

5.2 未来工作展望

本课题中依然有以下几方面的工作需要进一步研究讨论：

1) 在 web 浏览器访问虚拟机的工作中，可以设计更加人性化的界面以便于用户操作。可以考虑将虚拟机的操作界面嵌入浏览器页面中动态地显示。我在研究中可以实现了显示并以一定的频率刷新，但是不能对远程虚拟机进行操作，可以在后续工作中完成，这需要对浏览器语言具有较深的理解能力。

2) 在虚拟机的安全共享工作中，可以简化部署步骤，或者实现自动部署，减少用户和服务器维护者的工作量。

插图索引

表格索引

参考文献

[1] 吴吉义，平玲娣，潘雪增，等. 云计算从概念到平台[J]. 电信科学，2009 年，12 期：23-30

[2] Luiz Andre Barroso, Jeffrey Dean, Urs Holzle. Web search for a planet: the Google cluster architecture. In: IEEE Micro, 2003

[3] 王庆波，金淳，何乐，等．虚拟化与云计算[M]．北京：电子工业出版社，2009

[4] 邹大斌．桌面虚拟化将成今年热点[N]．计算机世界报，2010 年 2 月 1 日(第 34 版)．

[5] 马荟．虚拟桌面落云端[J]．互联网期刊，2010 年 5 月 10 日．

[6] 王春海．虚拟机深入应用实践[M]．北京：中国铁道出版社，2009: 9-10．

[7] Oracle VM VirtualBox User Manual. http://www.virtualbox.org/manual/ch08.html.

[8] Oracle VM VirtualBox User Manual. http://www.virtualbox.org/manual/ch08.html#id2643961.

[9] 李建设，吴庆波．基于 OpenSSL 的 VNC 安全性研究及实现[J]．微计算机信息，2005 年，33 期，6-9

[10] Oracle VM VirtualBox User Manual. http://www.virtualbox.org/manual/ch07.html

[11] rdesktop: A Remote Desktop Protocol Client. http://www.rdesktop.org/

[12] 杨训庭，jarsigner-JAR 签名和校验工具．http://blog.csdn.net/yangxt/archive/2007/09/23/1796965.aspx

[13] 金汉均，仲红，王双顶．VPN 虚拟专用网安全实践教程[M]．北京：清华大学出版社，2010:194-195

致 谢

首先衷心感谢曹军威研究员对我的悉心指导，曹老师帮助我选定课题内容，指明技术方向，在研究的过程中提出了很多非常宝贵的意见。在曹老师的关心和帮助下，我的毕业设计得以顺利完成。不仅于此，曹老师兢兢业业、勤勤恳恳的工作作风使我深有感触，受益匪浅。

同时我要衷心感谢万宇鑫师兄对我的热情帮助，他指导和督促我完成课题研究，指出我工作中存在的不足，帮助我不断完善课题方案。

我还要感谢陈伟同学，很多问题就是在与他的交流讨论中得以解决，他思路开阔，帮助我解决了部分技术难题。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：张瀚　　日　期：2010.7.1

附录 A 外文资料的书面翻译

虚拟云的安全信任模型

摘要—互联网云建立在多个数据中心之上提供服务。由于在服务商和云用户之间缺乏信任，云计算这一需求却不能被普遍地接受。云资源的伸缩性和庞大的数据处理受制于数据锁定，安全漏洞，隐私和版权。按需置备的云资源特别容易受到网络攻击。我们提出一个新的信任模型来集成虚拟集群和虚拟存储以便于可信赖的访问。为了保护云和站点级别的数据中心，我们设计出一个可信任的覆盖网络来完成声誉系统。我们可以通过在文件访问时的水印级别来保护数据。这里对各种实际生活中的云计算应用的安全对策进行评估：IaaS,PaaS 和 SaaS。

关键词—互联网云，数据中心，网络安全，虚拟化，信誉系统，信任模型，网络服务。

1. 导言

云计算通过配置硬件，软件，数据集动态地在虚拟的平台使用弹性资源。我们的想法是利用服务器集群和在数据中心的庞大的数据库，来实现桌面计算。云计算利用其低成本和简单来满足用户和提供商，虚拟化机器使这种成本效益变成可能。云计算在满足用户的许多不同种类的应用的同时，还要保证云生态系统的设计必须是安全，可信可靠的。

过去，eBay 和亚马逊已经开发出信任模型来保护电子商务和网上购物的用户。对于网络云服务，信任和安全变得更加苛刻，否则云服务提供商就要面对个人电脑和服务器用户的强烈抵制。如果缺乏隐私，安全和和版权保护，云平台会使用户担忧。作为一个虚拟环境，云提出了比传统客户端服务器端更加难以控制的安全威胁。为了解决这些信任问题，我们在数据水印的基础上提出了一个新的数据保护模型来保证安全。

虚拟资源和数据中心都面临着许多业务的不确定性。如何有效地管理这些不确定性是可信云计算中最困难的挑战。我们的工作是从先前的可信任模型中拓展，主要是由何等，李等，和宋等提出的点对点(P2P)网络和网格。我们的目标是确保云计算环境的稳定。基于信誉的 P2P 网络信任和社会网络需要重新设

计来确保数据中心和云平台。我们渴望能拥有一个健康的云环境，能够远离暴力，欺骗，黑客攻击，病毒，谣言，色情，痉挛，隐私和版权的侵犯行为。

其余各节的编排顺序如下：第 2 节我们先回顾云服务模式评估现有的云平台；然后第 3 节我们建议建立一个新的安全架构；第 4 节我们提出一个新的维护数据完整性和加强数据中心安全的信任模型；第 5 节我们提出通过综合信托管理数据访问保护的方法。最后，我们总结我们的成果，讨论进一步的研究。

2. 云服务模式和适用范围

我们评估已经投入使用的三种云服务模式的安全要求：IaaS,PaaS,和 SaaS。这些模型都是基于提供商和使用者之间的不同的服务水平协议(SLA)。图 1 显示了云模型采取不同的做法映射到不同的安全水平。表 1 反映了主要云提供者和他们服务反馈。

A. 三种云服务模式

面向虚拟基本设施的云服务(IaaS)：这种模型允许用户租用处理，存储，网络和其他资源。用户可以配置和运行客户机操作系统和应用程序。用户不用管理或者控制底层云基础设施，但是能够控制操作系统，存储器，应用程序，并且能够选择网络组件。亚马逊弹性云提供最多的 IaaS 服务。

面向平台的云服务(PaaS)：这种模型为用户在云基础设施上提供配置应用程序，这个设施是使用提供商供应的编程语言和软件工具建立的。用户不管理底层云基础设施。现阶段大多数供应商提供 PaaS 服务，除了亚马逊。

面向软件的云服务(SaaS)：这是指浏览器发起成千上万客户的应用程序。在客户端，不需要在服务器和软件许可上负担前期投资；在供应端，同传统的用户程序服务相比成本更低。谷歌，微软和 Salsforce.com 主推 SaaS 服务。

云服务提供四种配置模式：私有模式，公共模式，托管模式和混合模式。这些模式需要不同层次的安全措施。不同的 SLA 和服务的配置模式意味着这是云供应商，云消费者和第三方云软件供应商共同的责任。云服务的关键问题包括数据的完整性和保密性，以及供应商和用户之间的可信模式。

B. 云供应商和反馈服务

例如，谷歌拥有数以千计的超过 460000 台服务器的数据中心。该平台由服务器群集、谷歌文件系统和数据中心组成。在 2008 年，谷歌已经拥有 200 个这类群集。数据项存储在文本、图像和视频中。谷歌的 AppEngine(GAE)支持云和

web 应用程序。最好的 SaaS 应用程序是 IBM 的 Lotus Live、谷歌的 Gmail 和 Docs，以及 Saleforce.com 的在线顾客关系管理(CRM)。

目前有八个 IBM 研究中心支持 RC2(研究计算云)。IBM 的 BlueCloud 为云计算提供一个全面系统解决方案。这个系统销售整个服务集群加上开源软件，这些软件包括 Apache Hadoop 和 IBM 研发的资源管理和业绩鉴定软件包。亚马逊运行一个全球性的电子商务平台，为数以百万计的客户服务。亚马逊的弹性云由硬件和软件服务的灵活性决定。EC2 提供了一个运行虚拟服务器的环境。S3 提供无线的在线存储空间。亚马逊网络服务(AWS)提供 EC2 和 S3 服务。

3. 可信云/数据中心架构

风险云平台已使企业和政府丧失了数十亿美元。互联网云被认为是大规模的服务器集群。云平台由服务器、软件和数据库资源动态地配置形成。云服务器可以是物理机或者是虚拟机。用户界面接到请求服务器，请求服务器供给所需的工具。

A. 云担保平台

图 2 展示了一个有新的安全意识的云架构。云安全执行涉及许多方面。诸如蠕虫、病毒和 Dos 等恶意软件利用系统漏洞破坏系统功能或未经授权入侵访问关键信息。因此，我们需要安全防护来保护所有的群集服务器和数据中心。

- 保护服务器免受蠕虫、病毒等恶意软件攻击。
- 保护管理程序或虚拟机监控免受攻击。
- 保护虚拟机和监控免受破坏，拒绝服务攻击。
- 保护数据和信息免被窃取，侵吞。
- 提供认证和对紧要信息和服务的授权。

表 2 提出五项保护机制，来维护公共云和数据中心。恶意入侵可能会破坏宝贵的主机、网络和存储资源。一旦路由器、网关或分布式的主机发现异常，可能会停止云服务。通常情况下 SLA 层完成信任交流。公钥结构(PKI)的服务可以通过证书颁发机构(CA)的数据中心的信誉系统的辅助来推广。蠕虫和 Dos 的攻击必须加以制止。后面的章节我们进一步讨论信誉系统和文件访问控制。

B. 虚拟化云安全

虚拟化可以提升云的安全性。但是虚拟机增加了一个附加层，这就是容易变成单故障点的软件层。借助虚拟化，一台物理机器可以变成多个虚拟机(例如

服务器整合)。这样每一个虚拟分区可以更好地安全隔离，每个分区可以免受其他分区的 DoS 攻击。单独虚拟机受到攻击不会影响到其他虚拟机，出现故障也不会传播到其他虚拟机。系统管理程序监视每个客户系统，各个客户完全隔离。故障控制和故障隔离位虚拟机提供一个更加安全可靠的环境。

沙箱提供了一个安全的程序运行平台。此外，沙箱可以为客户操作系统提供一套严格控制的资源，使得客户操作系统允许定义一个安全的试验台来运行第三方未经测试的代码和程序。随着虚拟化，虚拟机从物理硬件中脱离出来。整个虚拟机可以表示为软件的一部分，可以当做二进制或者数字数据。虚拟机可以保存、复制、加密、移动、或者轻松恢复。虚拟机拥有更高的可用性和更快的恢复性。

我们建议使用虚拟机的实时迁移，这是专为构建分布式入侵检测系统(DIDS)设计。多个入侵检测系统虚拟机可以配置在各种资源站点，包括数据中心。DIDS 的设计需要 PKI 域之间的信任与否。安全策略的冲突必须在设定时得到解决并定期更新。我们需要防御计划来避免用户数据受到服务器攻击。用户的隐私资料不能泄露给其他未授权的用户。谷歌平台基本上采用内部软件来保护资源，亚马逊 EC2 则采用 HMEC 和 X.509 证书。

C. 云软件的保护

我们想要保护云环境中浏览器引出的应用程序。例如，一个 SaaS 从一个共同的云平台制订旅行或秘书服务。管理服务提供商在用户应用程序内协调服务和价格。用户可以在系统集成之前预见到配置。雅虎的 Pipes 是一个很好的轻量云平台。随着文件和数据的共享，隐私、安全和版权在云计算环境中可能会受到威胁。我们希望能够工作在一个这么一个软件环境中，这个环境可以提供许多基于大型数据集的工具来构建云应用。我们确定了以下几项所需的安全功能。

- 安全 web 技术全面支持的动态 web 服务
- MapReduce、BigTable、EC2 和 3S 对地理空间应用的个性化扩展
- 可实现快速查询和信息检索的稳定和持久的数据存储
- 验证用户和使用商用账户发送邮件的特殊的 API

D. 数据完整性和隐私保护

我们希望能够工作在一个这么一个软件环境中，这个环境可以提供许多基于大型数据集的工具来构建云应用。除了 MapReduce、BigTable、EC2、3S、

Hadoop、AWS、AppEngine 和 WebSphere2，我们发现以下一些云用户所需的安全和隐私功能。

a.验证用户和使用商用账户发送邮件的特殊的 API

b.依靠安全协议诸如 HTTPS 或 SSL 来访问云资源

c.细粒度访问控制，这样利于保护数据的完整性并阻止入侵或黑客

d.保护共享数据以免被恶意修改、删除或盗版

e.保护 ISP 或者云服务提供商(CSP)以免侵犯用户隐私

f.用户端个人防火墙

g.坚持执行云服务提供商的隐私政策，防止身份窃取，和 web 漏洞。

h.在 VPN 资源站点和目标数据之间使用 VPN(虚拟私人网络)通道。

4. 完整数据的信任模型

我们提出了一个新的信任模型来保持数据的完整性。该模型基于数据着色和云水印。我们还研究了数据锁定问题，并探讨其可行的解决方案。

A. 供应商与用户的责任

我们为大部分服务提供商和云用户确定了三个安全要求：保密性、完成性和可用性。图 3 所示，按照 SaaS、PaaS 和 IaaS 的顺序，提供商逐渐减小对云用户的安全控制。概括地说，SaaS 模式依赖供应商来执行所有的安全职能。而另一方面，IaaS 模式希望用户承担几乎所有的安全职能。PaaS 模式依赖供应商来维护数据的完整性和可用性，但是用户负责隐私保护和控制。

PaaS 模式依赖供应商来维护数据的完整性和可用性，但是需要用户自己保证数据隐私。我们列出了在动态云环境中影响数据安全的若干重要因素：保密性、数据完整性、访问控制、攻击防御、信誉系统、版权保护、数据锁定、应用程序编程接口、数据中心的安全策略、信任协商等。我们建议对数据中心的文件实行细粒度访问控制。为确保弹性资源的安全，信誉系统需要保障分散的资源站点和数据中心。还有必须维护站点的安全指标和用户访问记录。

B. 数据锁定问题和合理的对策

云计算将计算和数据都转移到供应商维护的服务器群。一旦数据移动到云中，用户不能方便地提取自己的数据和程序进而转移到其他服务器上。这就导致了数据锁定的问题。这阻碍了云计算的应用。数据锁定有以下两个原因：(1)

缺乏互操作性：每个云供应商都有自己专有的 API，这限制了用户提取数据；(2)应用程序兼容性：大多数云计算期望用户从头写新的程序。

对于数据锁定问题一种解决方案是使用标准化的 API。这就需要建立标准化的虚拟平台，坚持开放虚拟格式(OVF)—一个独立、搞笑、可扩展和开放格式的虚拟机平台。使用 OVF，用户可以数据从一个应用程序移动到另一个。这将提高服务质量，从而使得跨云应用变为可能。部署应用程序不需要重写每一个云，这样我们可以访问和集成不同的云服务。

C. 数据着色和云水印

信任是一个社会问题，而不是一个纯技术问题。然而，这个社会问题可以通过技术手段解决。我们相信技术可以在任何互联网应用中增加信任、公正、信誉、信用和保证。通过将云存储和水印相结合，云安全就是一个社会属性。这个概念在图 4(a)中有说明。数据着色意思是用特殊的颜色标识数据对象来将其分类。不同颜色的数据要经过不同的安全检查过滤。用户身份也被着色然后与数据相匹配，以启动不同的信任管理。云存储提供生成、嵌入和提取水印的过程，如图 4(b)所示。这种方法更多的细节可以再 Atallah 等，李等人和杨等人的工作中查到。

5. 数据中心的信誉向导保护

信任是个人的看法，这更主观。信任可以传递，但并不一定对称。信誉是一个公众的意见，这更加客观，而且往往依赖于广泛的民意汇集。信誉可能会随着时间变化或衰减。对比以前的信誉，近期的应该得到更高的关注。在本节中，我们通过信誉系统来保护数据中心或个人、团体云用户。边栏提供了信誉系统的设计方案。

信誉是公众对一个实体的角色或身份(如诚实或可靠性)的看法，这个实体可以是一个人，一个代理商，一个产品或者一项服务。它代表了一群人/代理商和资源拥有者的集体评价。在这之前很多系统已经引入了信誉机制，如 P2P，多代理和电子商务系统。我们使用系统方法来实现对云服务的信誉系统。

为了支持信任云服务，我们建议采用分布式信任和信誉系统。在黄等人的工作中，我们已经设计出一个基于 DHT 层次结构网络，来保证可信任的数据中心和应用程序。图 5 显示了支持更新的 web 和云服务安全基础设施。底层是信任协商和信誉集合的覆盖层。顶端是抵御病毒、蠕虫和 DDoS 攻击的安全覆盖层。我们的计划也防止网上盗版和版权侵犯。

该系统可以部署信任覆盖网络。我们建议使用 P2P 信誉系统层次结构来保护在文件级别的云资源。这就需要对共享资源实行粗粒度和系列度的访问控制。这些信任系统在各个级别跟踪保证安全。信誉系统的设计必须有利于云用户和数据中心双方。云计算中用到的数据对象存储在超过一个“存储区域网络”(SAN)。

数据的一致性需要在多个数据库中检查。版权保护保证了大范围的内容分布。为了将用户数据从专门的应用程序中分离出来，我们假定类似于 SaaS 的云应用，提供商有提供数据完整性和维护的责任。用户使用他们自己的数据切换不同的服务。只有拥有钥匙的用户才能访问到请求的数据。数据对象必须是唯一命名的，以确保全球的一致性。为了确保数据一致性，禁止未被授权的数据更新。

6. 结论和建议

本文介绍了一种新的信任模型和评估系统来建立云用户与数据中心之间的安全连接。我们的模型适用于公共和私有的云。该模型主要是保护虚拟化云基础设施。云平台的信任不仅基于服务水平协议，也来自于能够有效地执行安全策略以抵御网络攻击。通过不同的安全控制和信誉的标准，我们可以配合云的动态变化。我们的目标是建立一个可信赖的云环境，以保证高质量的服务和高保障的性能。

原文索引

[1] Kai Hwang, Sameer Kulkarni, Deyi Li. Trust and Reputation Modeling for Securing Virtualized Clouds over Datacenters. *IEEE Internet Computing*, Jan. 3, 2010.

综合论文训练记录表

学生姓名	钱瀚	学号	2006011551	班级	白 61
论文题目	云计算虚拟资源的安全访问与共享				
主要内容以及进度安排	3 月 1 号——3 月 21 号：调研，阅读文献。 3 月 22 号——4 月 4 号：虚拟机的动态配置，完成初步的客户端到服务器的访问。 4 月 5 号——4 月 25 号：浏览器到虚拟机服务器的访问，虚拟机动态响应，添加安全机制。 4 月 26 号——5 月 16 号：完成虚拟资源的共享模式。 5 月 17 号——6 月 6 号：撰写毕业论文。 6 月 7 号——答辩：准备答辩 指导教师签字： 考核组组长签字： 2010 年 3 月 26 日				
中期考核意见	钱瀚同学在调研和阅读文献的基础上对虚拟机的安全访问与共享进行了深入研究，但系统实现与搭建等方面的工作要抓紧。 考核组组长签字： 2010 年 4 月 28 日				

指导教师评语	钱翔同学的主要工作在于虚拟机的安全访问与登录，运用OpenVPN技术控制虚拟机的访问。钱翔同学的工作思路清楚，表达明确，希望在探索与创新能力等方面进一步提高。 指导教师签字： 2010年 6月29日
评阅教师评语	该课题主要研究了用户对虚拟资源的利用，对虚拟机的安全机制做出探讨，设计了虚拟机安全登录与使用的系统。在整个设计过程中，该同学基础扎实，思路明确，技术也达到课题要求。 评阅教师签字： 2010年 6月28日
答辩小组评语	论文工作选题需求明确、实用，工作量充分，完成课题目标；答辩表达流畅，思路清晰，回答问题正确。 同意通过论文答辩。 答辩小组组长签字： 2010年 6月29日

总成绩：85

开题88(15%)、中期82(25%)、答辩85(60%)

教学负责人签字：

2010年 7月 1日