分类号：TP311.5　　单位代码：10335

密　级：无　　学　　号：21451164

浙江大学

硕士学位论文

中文论文题目：基于 RHEV 虚拟化平台的自动化管理系统的设计与实现

英文论文题目：Design and Implementation of Automatic Management System Based on RHEV Virtualization Platform

申请人姓名：陈家星

指导教师：周　波 教授

合作导师：尹可挺 博士

专业学位类别：工程硕士

专业学位领域：软件工程

所在学院：软件学院

论文提交日期 2016 年 01 月 05 日

基于 RHEV 虚拟化平台的自动化管理系统的设计与实现

陈家星

浙江大学

基于 RHEV 虚拟化平台的自动化管理系统的设计与实现

论文作者签名:_____

指导教师签名:_____

论文评阅人 1: _____

评阅人 2: _____

评阅人 3: _____

评阅人 4: _____

评阅人 5: _____

答辩委员会主席: _____

委员 1: _____

委员 2: _____

委员 3: _____

委员 4: _____

委员 5: _____

答辩日期: _____

Design and Implementation of Automatic Management System Based on RHEV Virtualization Platform

Author's signature:_____

Supervisor's signature:_____

Thesis reviewer 1: _____

Thesis reviewer 2: _____

Thesis reviewer 3: _____

Thesis reviewer 4: _____

Thesis reviewer 5: _____

Chair: _____
(Committee of oral defence)

Committeeman 1: _____

Committeeman 2: _____

Committeeman 3: _____

Committeeman 4: _____

Committeeman 5: _____

Date of oral defence: _____

浙江大学研究生学位论文独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 **浙江大学** 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名: 签字日期: 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解 **浙江大学** 有权保留并向国家有关部门或机构送交本论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 **浙江大学** 可以将学位论文的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名: 导师签名:

签字日期: 年 月 日 签字日期: 年 月 日

摘要

随着虚拟化技术的发展，越来越多的企业投入到了虚拟化平台开发和应用。RHEV 是由小红帽开发的极其优秀的虚拟化平台，其高可用性，高扩展性，高性能，操作性简易等优点使 RHEV 得到了业界一致好评。本文的重点在于基于 RHEV 虚拟化平台的自动化管理系统设计与实现，首先根据用户的要求，系统负载均衡以及站点崩溃后恢复重建的需要，明确自动化管理系统需要对虚拟机 VM（Virtual Machine）、模板（Template）、虚拟机池（VM Pool）、模板迁移（Template Migration）等操作功能的实现。根据自动化管理系统的设计将整个系统设计成三层结构，即前端页面、中间业务处理层、后端任务调度执行层。在前端页面进行了数据优化和数据筛选过滤，将有效数据传入中间业务调度层。中间业务调度层根据不同的业务以 RPC 的方式请求服务端获得数据，将用户数据和业务必要信息包装在一起提交给中间业务处理模块创建任务。由后端任务调度执行模块负责任务的调度和任务的执行。从而实现 VM 的创建，VM Pool 的创建，模板的创建，模板的迁移等功能。本自动化管理系统经过运行测试，实现了对 RHEVM 中 VM，模板，VM Pool 的自动化管理和两站点之间模板对自动化迁移，并能保证其可用性和稳定性。

关键词： 虚拟化技术，RHEV，自动化管理 ，VM Pool，模板迁移

Abstract

With the development of virtualization technology, more and more enterprises work on the virtualization platform development and application. RHEV is an excellent virtualization platform developed by Red Hat. RHEV gets good reputation in IT field because of its high availability, scalability, high performance and easy operation. This paper focuses on the design and implementation of automation management system based on RHEV platform. Firstly, according to the requirements of users, the need of system load balancing and site reconstruction, the paper should implement the virtual machine VM, VM Pool, Template Migration and other operational functions. Automated management system will be designed into a three-tier structure of the system, the front page layer, intermediate business scheduling layer, back-end tasks layer. In the front page after the data optimization and data filter, the effective data will be transferred into the intermediate business scheduling layer. The intermediate service scheduling layer send the server requests to obtain data with different services in RPC mode, and then wraps the user data and business necessary information together to submit it to the back-end to create corresponding task. By the back-end task processing module is responsible for task scheduling and task execution. In order to achieve the creation of VM, VM Pool to create, Template to create, Template Migration and other functions. The automation management system has been running and tested to realize the automatic management of VM, Template and VM Pool in RHEVM and the automatic Template of Template between the two sites, and can guarantee the usability and stability of these functions.

Key Words: Virtualization technology, RHEV, automated-management, VM Pool, Template Migration

目录

图目录

表目录

第1章 绪论

1.1 选题背景及研究内容、意义

20 世纪 90 代，计算机因为 Windows 这一优秀的桌面操作系统，开始向个人电脑时代迈进。个人计算的发展也使得整个计算机行业的一致认可了 intel 公司设计 x86 架构。同时企业服务器操作系统 Unix／Linux 对 X86 的支持，使得 X86 一度称为整个计算机行业的设计标准[1]。也恰恰因为这一统一的标准，互联网和个人电脑得到了急速的发展和广泛的普及。近年来随着今年来物联网、Web 应用、移动互联网、互联网+等产业的迅猛发展。面对以几何倍数激增的数据量和业务量，企业不得不增添员工和服务器数量缓解这样的压力。服务器不断增多也为企业服务器部署增加了许多困难，为企业管理人员带来了诸多难题。硬件利用率低，架构复杂，维护成本高，IT 管理成本激增[2]，安全性得不到保障，关键性应用面对故障脆弱，数据灾后重建困难等等问题，虽然企业花费很大力气进行解决，但都收效甚微。有关数据显示[3~5],许多企业数据中心的 CPU 的利用率不超过 25%，内存利用率不超过 65%。这表明按照目前的资源分配方式,硬件资源利用率较低，用户的总体拥有成本（Total Cost of Ownership，TCO）相对较高[6]。硬件资源的不合理利用不仅造成了很大浪费，还会造成企业服务质量下降。X86 虚拟化技术通过将基于 X86 架构的独占型硬件资源转变为更为通用的共享型硬件，最大限度的提高硬件的利用率。虚拟化技术将计算机的各种资源（CPU，内存，网络，硬盘空间，网卡等）予以抽象、转化成可以分割、组合而又彼此独立的一个或多个计算机配置环境[7~8]。充分挖掘硬件的潜在能力，大大提高硬件利用率，降低硬件维护和运营的成本，简化运维方式，动态迁移技术的应用，实现了硬件分配的均衡[9~10]。虚拟化技术不仅为企业节约更多时间，并使成本尽可能多的投入到业务上。

本课题是本人实习的过程中接手的虚拟化桌面自动化（VDI_AUTOMATION）中的子项目。在实现这个项目，了解了虚拟化，集群，模板，虚拟机，虚拟池等一系列的概念。虚拟化已经在本人实习的公司广泛应用。在本人学习和应用虚拟化过程中深刻认识到这项技术在未来可能会对我们的生活带来一系列深刻的影响。希望通过这篇论文是更多的企业和个人进一步认识虚拟化技术，进一步的投入研究、开发、应用虚拟化技术。希望在不久的将来能有一个以虚拟化技术为核心构建统一的，协调的，完备的，全方位的系统架构。在便宜生活的同时也能推动社会的进步这也将是我写这边文章最大的期望。

1.2 虚拟化技术发展的国内外现状

虚拟化技术由起源于 20 世纪 60 年代[11]，兴起于 70 年代虚拟化监控器（Virtual Machine Monitor）发展起来的一项技术[12]。此项技术由 IBM 公司开发应用在大型机监控系统软件，它允许在已有的物理硬件的上生成许多可以独立运行的小型操作系统（在虚拟化环境中称之为虚拟机）。随着这种虚拟化技术的问世，各大服务提供商找到了一种解决硬件资源利用率低与企业巨大数据量、繁重的业务需求之

间的矛盾的办法。在各大服务商的竞争下，虚拟化技术得到了巨大的发展，并且由原本的只能运行在大型机向可以在小型机和 Unix/Linux 服务器上运行转变[13]。之后这种技术被各大服务器提供商应用在了高端服务器系统中。随着虚拟化技术的不断应用和企业对其效用的不断的认可，在之后的几十年中虚拟化技术不断发展成熟。形成了包括硬件虚拟化、指令集虚拟化、操作系统虚拟化、应用软件的虚拟化等一系列完整的虚拟化解决方案。虚拟机技术的发展特别是 X86 架构的虚拟化技术的发展，极大的推动力虚拟化技术在 IT 行业的普及和应用。面对软件方面的虚拟化发展到一定的技术瓶颈。Intel 公司在 2005 年推出了支持虚拟化技术的 CPU 产品线，并开始推广应用 Intel VT（Intel Virtualization Technology） [14]。几乎同一年，AMD 公司也发布了支持 AMD VT（AMD Virtualization Technology）的一系列处理器产品。随之往后两家公司每代产品中都支持虚拟化技术。与此同时，在软件方面 VMware 推出了支持实时迁移的 vSphere 4.0，VMware 虚拟机也凭借其系统的高可用性、实时迁移、安全稳定等的特性在市场一直保持竞争优势 。Microsoft 公司也随之推出了 Hyper-V2.0，其优秀的虚拟化性能也得到了业界的认可[15~16]。与这两家开发的桌面虚拟化产品相比，思杰开发的应用型虚拟化产品应用更为普及。思杰公司也因产品的可管理性、高安全性、高兼容性以及有效降低 TCO 具有很高的市场占有率。红帽公司推出的开源的平台虚拟化产品 Ovirt 和企业级虚拟化产品 RHEV 也部署简单、安全性好、可维护性强、技术支持完备被用户所喜爱。

与国外相比，国内的虚拟化技术起步较晚，主要更侧重在对虚拟机的应用。在国外已将虚拟机技术应用到电信、金融等行业的时期[17~18]，国内对其仍保持了解观望态度。之后虚拟化技术在国外成功应用和广泛普及，国内才对这一技术开始看中。近些年随着大多数系统服务商的推动，虚拟化技术在国内已经运用到了科研，教学和商业中。其中华为公司在 2014 年推出起 Fusion Sphere 5.0 的虚拟化解决方案，阿里巴巴公司也在早年推出了基于虚拟化技术的阿里云服务，腾讯公司的腾云 SDN 和百度公司的 Terminator 服务都是由国内大型公司推出的优秀虚拟化服务。

虚拟化技术因其高扩展性、高可用性、对软硬件资源的动态分配和负载均衡、提高系统资源的利用率、增加系统的维护性、降低系统管理成本等优势被各大应用提供商看重[19~21]。目前世界上主流的虚拟化解决方案有 XEN[18]，KVM，Hyper-V，VMware 等，各大厂商如 Amazon、IBM、Microsoft、VMware、CSICO 以及国内的阿里巴巴、华为、百度、腾讯等著名企业都将这种技术应用的自己的系统解决方案中并对外提供服务。

1.3 本文内容结构

本论文主要内容设计实现能够一个自动管理 RHEV（Red Hat Enterprise Virtualization）虚拟化平台中的各种虚拟化资源的自动化管理系统。本自动化管理系统是基于 RHEV 虚拟化平台进行开发的，依靠 RHEV 提供的 REST API 实现对虚拟机、虚拟机池、模板、网络、主机、存储、数据中心等高效率、自动化、高可用的管理系统。详细的工作内容如下:

1、自动化创建虚拟机，用户只需要输入虚拟机名称，选择要使用模板，Cluster 等必要信息系统会自动创建虚拟机，并执行 PowerShell Script 为安装 OS 作好准备。同时记录保存其日志文件。
2、自动化创建虚拟机池，提供的虚拟机池的名称，使用的模板，池的大小，池的描述，用户和域，角色和权限等信息系统自动为创建虚拟机池。同时记录保存其日志文件。
3、自动化创建模板，用户选择源虚拟机的名称，操作系统，虚拟机状态，用户和所在域，集群，角色和权限等信息系统自动为创建虚拟机模板。同时记录保存其日志文件。
4、自动化迁移模板，用户选择源站点，源站点中可用的模板并选择模板迁移的目标站点，设定模板在目标站点使用的 Cluster，Storage 等信息，系统自动迁移模板。
5、查看 VM、VM Pool、模板、模板迁移的基本信息和日志。查看其配置的网络，集群，存储信息、网络节点信息、用户组和域名、角色和权限等信息。
6、创建定时任务根据用户配置的未来时间点，现将任务写入结构化数据库中，定期调用查看任务是否需要执行，当时间点到达时启动任务，并记录任务执行和错误日志

1.4 本文基本框架

本文主要是设计实现基于 RHEV 虚拟化平台进行开发的，依靠 RHEV 提供的 REST API 实现对虚拟机、虚拟机池、模板、网络、主机、存储、数据中心等高效率、自动化、高可用的管理系统。在自动化管理系统设计和实现过程中我所做的工作是实现页面展示层和后端任务调度层之间的业务逻辑处理层。用户在前台页面输入必要信息、提交个人请求，请求经过中间业务逻辑处理之后生成相应的任务并提交给后端任务调度逻辑，调度进程会在后台进行执行任务，并把任务执行的结果返回到前端页面，展示给用户，同时写入日志。本文基本框架如下:

第一章绪论主要介绍本文的选题背景和虚拟化技术国内外的现状 以及自动化管理系统研究的内容和行文框架。

第二章自动化管理系统设计技术的基本原理主要阐述本系统会涉及到的技术难点如 RPC，KVM 架构，LDAP，RHEV 的负载均衡策略，实时迁移策略等。

第三章自动化管理系统需求分析，主要介绍本系统需要解决问题和本系统需要做哪些工作，并分析了虚拟机的创建、虚拟机池、模板的创建、模板的迁移的需求。

第四章自动化管理系统设计和实现，根据第三章的需求分析设计自动化管理系统怎么实现各个功能。分为自动化管理系统整体设计和关键技术设计实现。

第五章系统部署和测试，自动化管理系统部署需要的软硬件环境以及系统运行的截图，同时查看任务执行 RHEVM 中结果确保任务正常执行。

第六章总结和展望，主要对本论文所做工作的总结和虚拟机化技术未来发展的展望。在总结中，指出本文一些不足之处和改进办法。最后是致谢、参考文献。

1.5 本章小结

本章主要分为四个小节，分别讲了论文的选题背景和研究内容、虚拟化技术的国内外发展状况、本文内容和本文基本框架。

第2章 自动化管理系统使用的技术介绍

2.1 虚拟化技术

虚拟化技术按抽象程度的递增[21-22]的顺序被划分以下五个层次。

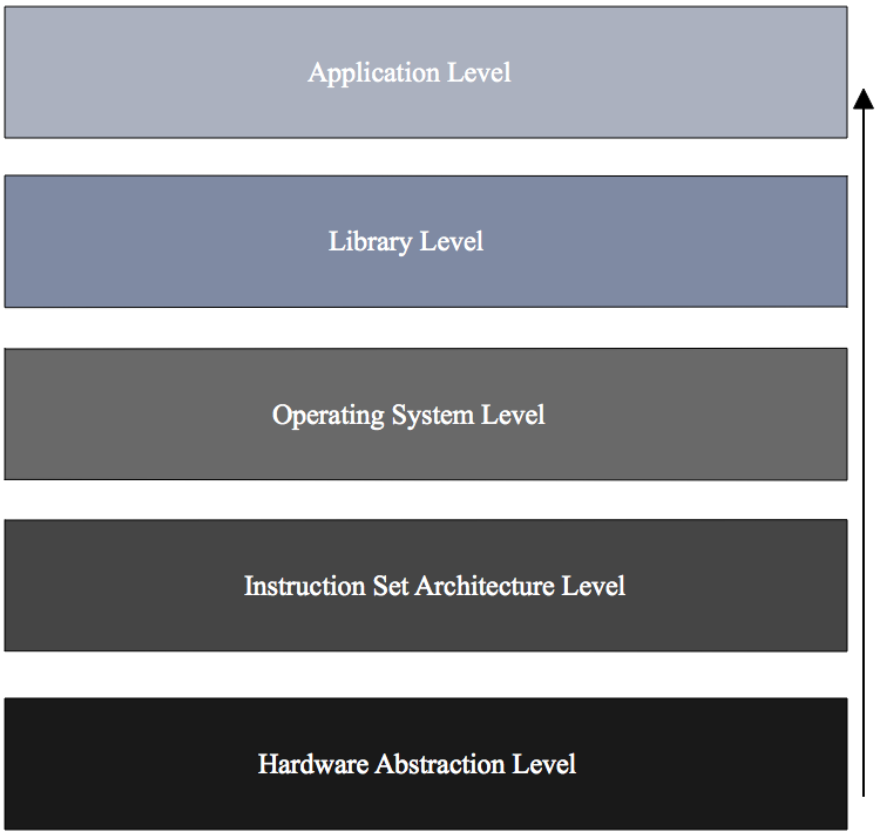

图 2.1 虚拟技术按的划分

2.1.1 硬件抽象层（Hardware Abstraction Level）

硬件抽象层虚拟技术[23]是指硬件虚拟技术，如 2005 年 Intel 公司推出的基于 x86 的硬件辅助虚拟化技术，而 2006 年 AMD 公司也推出了支持 x86 架构处理器的虚拟技术支持[24]。这两家公司推出的基于 x86 的硬件辅助虚拟化产品，在很大程度上提高了虚拟机性能，从而推进了虚拟化技术的进一步发展。

2.1.2 指令集结构层（Instruction Set Architecture Level）

指令集结构层虚拟技术是指以纯软件方式模拟 cpu 指令的执行的虚拟化技术。VMM 截获 Guest OS 发出的指令[25]，对把指令进行模拟、翻译成底层上可以运行的 CPU 内部命令或 IO 指令。

2.1.3 操作系统层（OS Level）

操作系统层虚拟技术是指在主机上操作系统上加入虚拟化的虚拟化技术。这种技术不需要 Hypervisor，虚拟机之间的硬件分配和虚拟机的隔离等工作由主机操作系统来负责。这种技术要求用户使用经过修改操作系统，并且虚拟机不允许运行多个操作系统。灵活性较差，性能接近裸机操作系统。

2.1.4 库层（Library Level）

库层虚拟技术是在操作系统和应用程序之间提供仿真接口的虚拟化技术。这种技术使只能运行在特定的操作系统的应用程序，能够运行在其他操作系统上。如 Linux 下的 Wine，可以让只能在 Windows 下才能正常运行的应用程序在 Linux 系统环境中也能正常运行。

2.1.5 应用层（Application Level）

应用层的虚拟技术是指能虚拟化应用程序的虚拟化技术[26]，如 Java 虚拟机、微软.NETCLI。

2.2 X86 不同的虚拟化

2.2.1 完全虚拟化

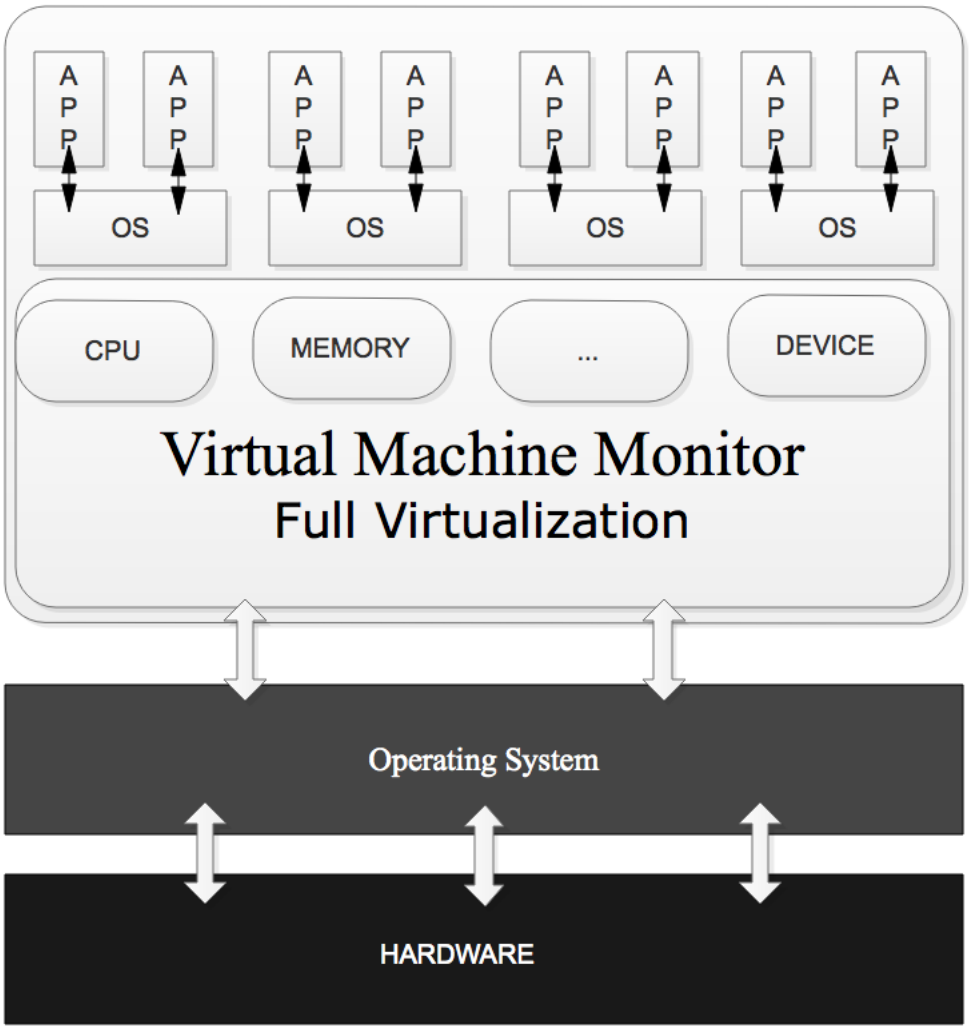

图 2.2 完全虚拟化结构图

完全虚拟化技术（Full Virtualization）[27]，又称原始虚拟化技术或是宿主型虚拟化。即是在硬件层和宿主操作系统层面上加入一层 Virtual Machine Monitor 层的虚拟化技术。Guest OS 利用主机操作系统提供的设备驱动和底层服务对底层硬件进行访问，来实现内存的管理，资源管理和进程的调度。使用这种虚拟化技术，用户使用不必进行修改的操作系统作为 Guest OS，对于 Guest OS 能够正常的访问底层硬件资源，无法感知运行在虚拟化环境中，这种虚拟化中虚拟机[10]的应用程序调用硬件资源时需要经过:虚拟机内核->VMM->主机内核->硬件资源。其原理是虚拟机系统调用 CPU 特权时采用二进制模拟/翻译的方法，将虚拟机系统的指令翻译成主机操作系统的指令进行调用底层指令。这样做的无疑造成较大系统

开销，据统计使用这种虚拟机的虚拟机运行性能比单纯的运行主机操作性能相比大约下降 10%～30%左右。因为这种虚拟化技术能够屏蔽底层硬件，完全依赖宿主操作系统，具有良好的兼容性[11]。宿主型虚拟化技术代表是 Workstation 和 Microsoft Virtual PC 、 Virtual Server 等。

2.2.2 半虚拟化

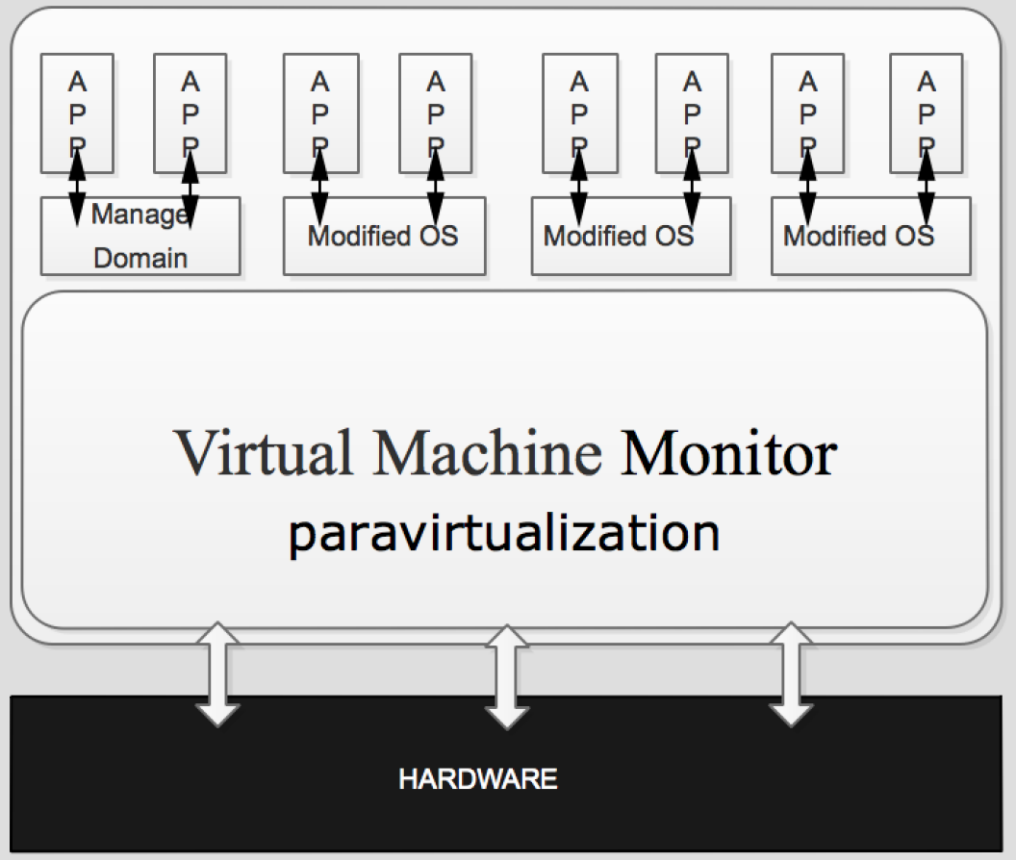

图 2.3 半虚拟化结构图

半虚拟化（Para-Virtualization），又称超虚拟化、部分虚拟化或是裸金属型虚拟化[28]。这种虚拟化技术 VMM 与硬件之间不需要主机操作系统，直接使用和管理底层的硬件资源。Guest OS 对硬件资源的访问通过 VMM 来完成，VMM 作为底层硬件的直接操作者，拥有硬件的驱动程序。这种虚拟化技术中的 VMM 直接管理调用硬件资源，有人也将半虚拟化中的 VMM 看作一个小型的操作系统。这种虚拟化技术使用 Guest OS 需要对原有操作系统进行特定的修改。修改后的操作系统相互独立，各自负责各自的虚拟机。因为这种方式 Guest OS 与 VMM 协作完成底层硬件的调用，不需要特定硬件支持，从而能达到裸机操作系统 97%以上性能[29]。因为这种虚拟化技术需要对源操作系统进行特定的修改，对于开源操作系统有较好的兼容性，对非开源的操作系统兼容性较差，甚至完全不兼容。代表是 VMware ESX Server、Citrix Xen Server、Microsoft Hyper-V、Linux KVM。

2.2.3 预虚拟化

预虚拟化技术[30]（pre-virtualization）由 Karlsruhe 大学、新南威尔士大学和 IBM 的研究人员共同提出一项虚拟化技术。这种虚拟化是指将源操作系统的特权指令设计成虚拟层可以调用的静态接口[31]。使用这种虚拟化的好处在于不需要对源操作系统进行特定的修改就可让 Guest OS 支持虚拟化。

2.3 虚拟机迁移技

虚拟机迁移是将原有主机（源主机）上的配置环境、主机状态、数据资源或其他资源迁移到用户指定的主机(目标主机)，并且保证迁移到目标主机上一切资源都能够正常使用或运行的过程[32]。迁移过程大致如下 1、在源主机上备份需要迁移的各种资源；2、将备份的需迁移的资源存储在存储介质中；3、目标主机从存储介质中读取迁移资源，并恢复到迁移资源备份时的状态。虚拟机迁移的目的使系统维护简单，提高负载均衡，增强错误容忍度和优化电源管理等[33]。

虚拟机迁移的性能可以根据整体迁移时间、停机时间、对目标主机上的应用程序的性能影响三个指标进行衡量[34]。从三个性能指标的角度老说，虚拟机迁移目标: 尽可能减小的整体迁移时间，停机时间最短(理想情况是不停机进行迁移)，对运行在目标主机上的应用的影响最少。

2.3.1 虚拟机迁移的分类[35]

1、物理机到虚拟机的迁移，简称 P2V（Physical-to-Virtual）。是指物理服务器上的操作系统、应用软件、事系统数据或事系统状态信息等资源迁移到 VMM 上的过程。在迁移物理服务器上的系统变量和存储数据等资源过程中会使用各种不同的迁移工具。迁移完成以后，目标主机会进行安装相应的驱动程序，并为其分配与源主机相同的地址（如 TCP/IP 地址等），在这些操作完成之后目标主机重启迁移过来的服务，代替源主机上的服务[36]。这种迁移方法分手动迁移和半自动迁移两种方式。它支持进行热迁移(即迁移过程中不宕机)，热迁移只能在源主机操作系统 Windows 的情况进行，对于源主机是其他操作系统的情况不支持。

2、虚拟机到虚拟机的迁移，简称 V2V（Virtual-to-Virtual）。是指源主机上的操作系统、应用软件、事系统数据或事系统状态信息等资源迁移目标主机上的过程。这种迁移方法的难点在于迁移过程中需要考虑主机和虚拟硬件上的差异。虚拟机从一个 VMM 迁移到另一个 VMM。这两个 VMM 的类型可以相同，也可以不同。如可以是 Xen 迁移到 KVM，也可以是 KVM 迁移到 KVM。依靠这种迁移技术将各种资源从源 Host 系统迁移到目标 Host 系统可以通过多种方式来实现。

3、离线迁移（offline 迁移），也称常规迁移或静态迁移。是指在迁移之前现将源主机暂停，如果是源主机与目标主机共享存储的情况，只拷贝源主机的系统状态信息到目标主机中，之后目标主机根据迁移过来的系统状态信息上重新设置系统运行状态并恢复各种服务的运行。如果是源主机和目标主机使用的非共享存储，则拷贝系统状态信息、虚拟机镜像及其状态到目标主机中。这种迁移方式在迁移过程中需要停止当前主机运行的各种服务而全力进行迁移。用户能明显感到迁移的这段时间内主机上的各种服务都不能使用。

4、在线迁移（online 迁移），也称实时迁移(live 迁移)[37]。是指在迁移的过程中虚拟机上的各种服务能够正常运行。其实迁移步骤与离线迁移大致相同。用户在迁移过程中几乎感觉不到服务的暂停，因为停机时间很短，这样能保证在迁移过程中虚拟机服务能够正常运行[38]。这种迁移方法是在前期准备阶段，源主机上的各种服务先运行到特定的阶段，目标主机将运行各种服务的必备资源迁移过来，并启动该服务，源主机将对该服务的控制权转移给目标主机，转移过程非常短暂，此时此服务就能在目标主机继续运行。因为切换服务的过程非常短暂，用户甚至无法察觉。这种迁移方式对用户是透明的，他适用与对服务可用性要求很高的场景。

5、内存迁移技术。XEN 和 KVM 的内存迁移都采用了一种由 Clark 和 Nelson[19] 等人提出预拷贝（pre-copy）技术[39]。预拷贝是对了对于 VM 的内存状态的迁移。它侧重点使停机时间缩减到最短。预拷贝可分为如下几个阶段 [40]:

1) Push 阶段: 在迁移开始之后源主机始终运行，在第一轮循环内，系统将源主机的内存数据拷贝到目标主机中，在第一轮循环周期拷贝所有内存页上的数据，之后每轮循环周期只拷贝被修改的脏页内存（dirty pages）。从而保证了源主机和目标主机内存中数据的一致性。

2) Stop-and-Copy 阶段: 暂停源主机的运行，内存不再进行更新，将源主机上的内存页面拷贝到目标主机机，拷贝完成后开启目标主机。

3) Pull 阶段: 此阶段是指在拷贝内存也的工程中，目标主机发先缺少某一页，向源主机要求重新传送该页面。

预拷贝技术因为尽可能传输最少的内存页，达到了尽可能少传输数据的目的，从而停机时间降到了最低。这种技术对于内存中更新速度非常快的，每次循环过程中内存都会变脏的内存页，致使 pre-copy 的循环次数非常多，迁移时间无可避免的被拖长的情况效果不是很理想。KVM 为了解决上述这种情况，对预拷贝制定了如下优化原则: 1)、每个循环周期内的 dirty pages 不超过 50，称为集中原则；2)、每个循环周期内传输的 dirty pages 不多于新产生的 dirty pages，成为不扩散原则；3)、循环次数不超过 30，若超过终止此次循环，称为有限循环原则。

根据以上优化原则，对每轮 pre-copy 的循环次数和效果进行计算，若 pre-copy 对于减少内存页效果不显著或者循环次数超过了上限的情况，中止此次循环，进入 Stop-and-Copy 阶段。pre-copy 过程和调度虚拟机进程运行的期间，执行内存的写操作不超过 40 次。这样直接限制了 pre-copy 的内存页变脏的速度，这样做的代价是限制虚拟机中进程运行。

2.4 RPC

RPC，即是 Remote Process Call[22]，即远程过程调用。它广泛应用在分布式系统应用程序之间的通信，是一种分布式应用系统之间通信的重要机制，也是一种于远程计算机程序交互的网络协议。它是分布式计算技术的基础。RPC 协议依赖与网络传输协议（如 TCP 协议或 UDP 协议）为通信程序之间提供数据的交互 [23]。, RPC 跨越了 OSI 网络通信模型中传输层和应用层。RPC 将原来的本地调用转变调用远程服务器上的方法，给系统等处理能力和吞吐量带来机会无限大提升。

RPC 的实现包括客户端和服务端，即客户端调用方和服务端调用方，服务调用方发送 RPC 请求到服务提供方，服务提供方根据调用方提供的参数执行请求的方法并返回结果，一次 RPC 调用完成。在调用方发起请求和服务方提供方执行完请求返回结果的过程中涉及的技术有数据的序列化和反序列化。它实现方式较多，技术比较成熟。其在 Windows 下等工作原理[24]。

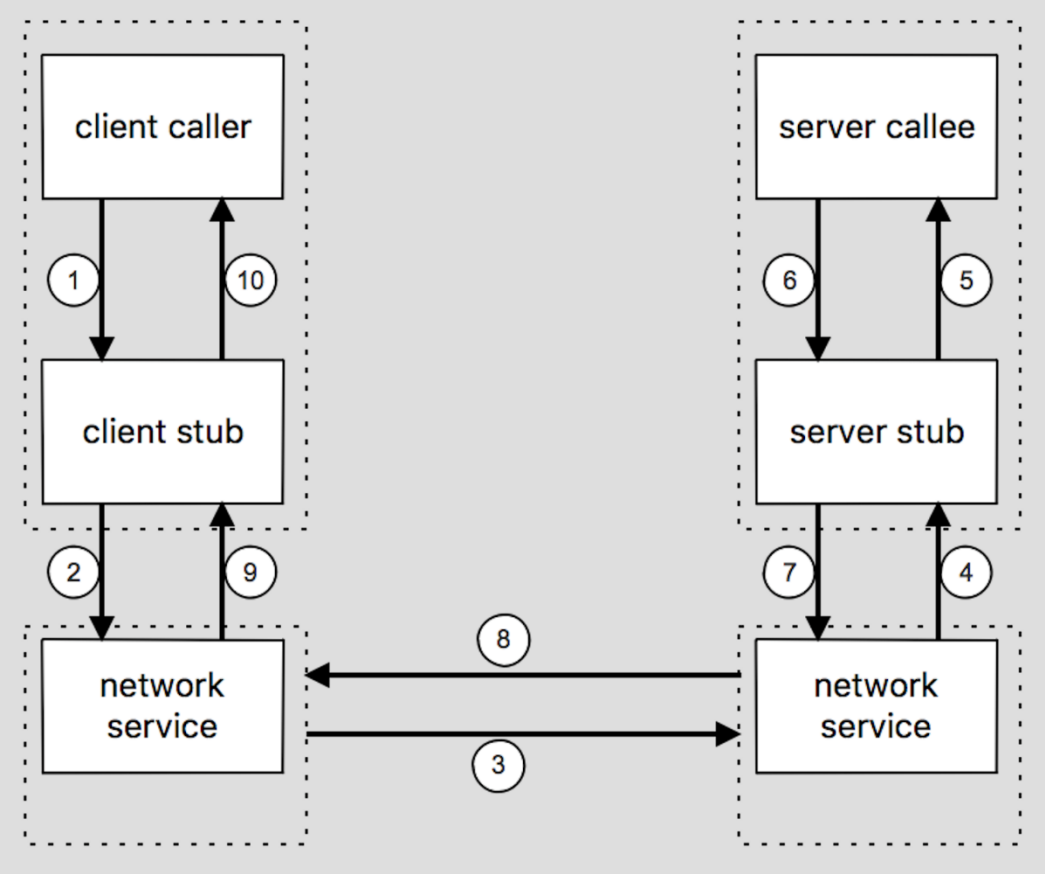

图 2.4 RPC 工作原理

客户机和服务器之间进行一次 RPC 交互过程，内部操作步骤如下:

1）客户端以本地调用方式调用句柄，传送参数。
2）调用本地系统内核发送网络消息。
3）消息传送到远程主机。
4）服务器句柄得到消息和参数，并进行解码。
5）执行远程过程。
6）本地服务执行将结果返回服务器句柄。
7）服务器打包返回结果，调用远程系统。
8）消息传回本地主机。
9）客户句柄由内核接收消息。
10）客户接收句柄返回的数据。

2.5 RHEV

RHEV，即 Red Hat Enterprise Virtualization[25]的缩写，中文全称红帽企业虚拟化。RHEV 是红帽公司推出的一款功能强大、性能稳定企业级别的服务器虚拟化平台。

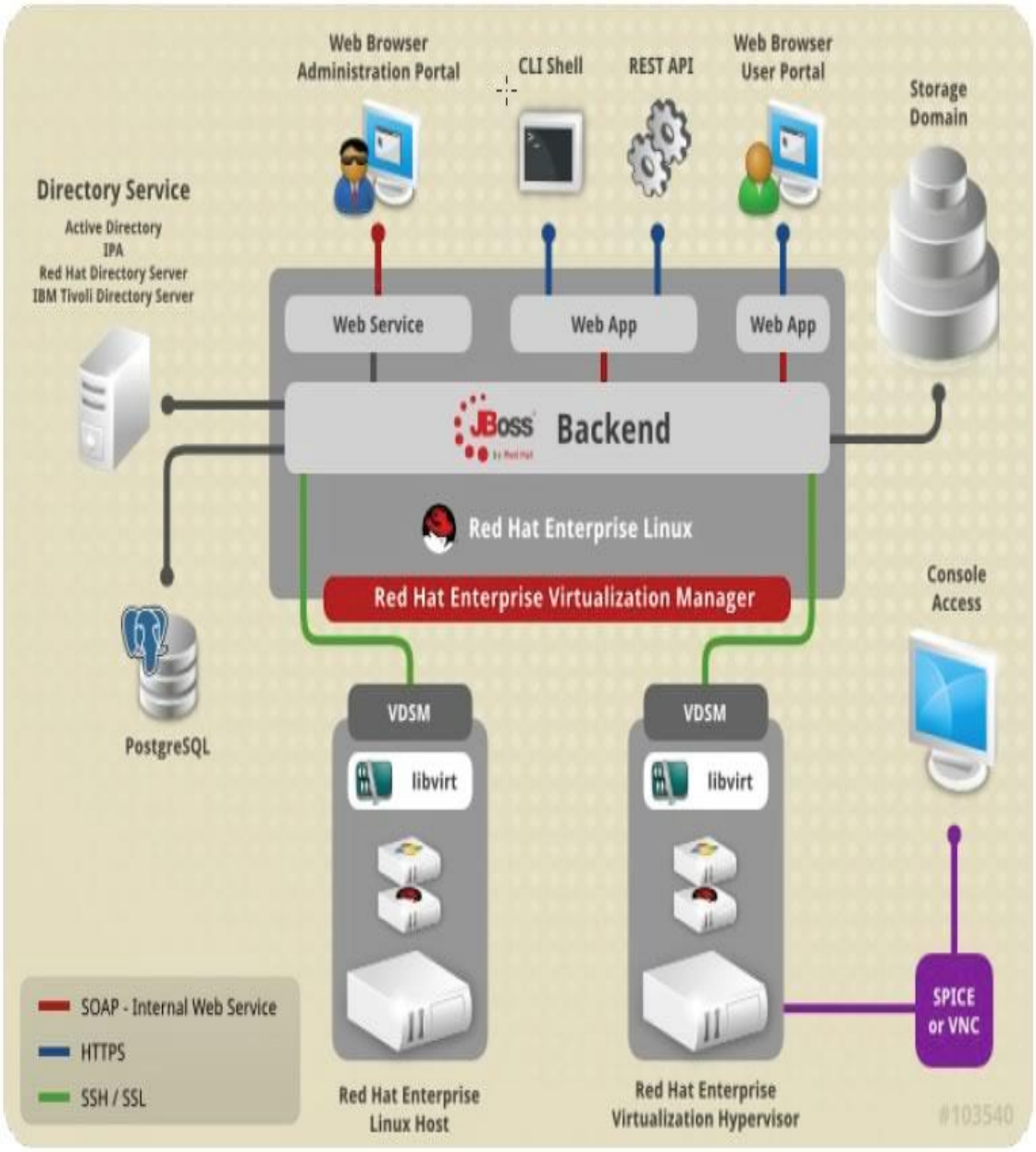

图 2.5 RHEV 的架构图

2.5.1 RHEV 逻辑资源

RHEV 的环境包括一个或多个主机（Red Hat Enterprise Linux 6.5/6.6，或 Red Hat Enterprise Linux 7 主机；或 RHEV Hypervisor 6.5/6.6，或 RHEV Hypervisor 7 主机），和至少一个 RHEV Manager。主机使用 KVM （Kernel-based Virtual Machine）虚拟技术运行虚拟机。

RHEV Manager 运行于 Red Hat Enterprise Linux 6.5 或 6.6 服务器上，对虚拟机、虚拟机镜像、存储配置、用户会话、连接协议、集群 、模板、数据中心、虚拟机池等资源进行管理[41-42]。同时，它也作为一个控制 RHEV 环境的接口，供用户调用。RHEV 系统的资源可以分为两类：物理资源和逻辑资源。

数据中心（Datacenter）是 RHEV 环境中的最高级的抽象概念，它被 RHEV 设定为包含虚拟环境中所有物理和逻辑资源一个大的容器。

集群（Cluster）由多个物理主机组成，为虚拟机提供各种资源的资源池。一个集群中等所有主机共享此集群中的网络和存储设备。集群可以看作虚拟机迁移域，即是在这个集群中的虚拟机可以随意迁移。

逻辑网络（Network）是对物理网络的逻辑抽象。VMM、存储设备、虚拟机、主机之间的网络流量被逻辑网络分成不同的组。

主机（Host）是物理的服务器在 RHEV 中的抽象。每个 host 中可以运行多组虚拟机。主机按照一定的逻辑划分成多个集群，同一个集群中虚拟机可以迁移。

虚拟机（VM）是虚拟化环境中对操作系统和操作系统上运行的应用程序的抽象。根据不同的用途可以分为虚拟台式机（Virtual Desktop）和虚拟服务器（Virtual Server）。虚拟机是用户管理的基本对象。一般用户拥有访问虚拟机的权限，管理员用户拥有创建、管理和删除虚拟机的权限。

模板是虚拟机和虚拟机配置信息的抽象。创建虚拟机最快的方法是利用模板进行创建，同一个模板创建出的虚拟机配置相同。

虚拟机池（VM Pool）是根据同一个模板创建出来的虚拟机的抽象概念 。

2.5.2 RHEVM (Red Hat Enterprise Virtualization Manager)

RHEVM 能管理虚拟环境中各种资源并能对这些资源进行功能性操作（创建模板、分配网络、安装 ISO 等操作），也可以管理两种类型的 Hypervisor。RHEV 默认带有 VMM，它是基于 RHEL 与 KVM 虚拟化，用于管理的物理节点。RHEL 上的虚拟机必须在 RHEVM 中进行注册，RHEVM 才能对其进行管理。

每个 RHEVM 最多管理 500 个可以和 RHEVM 连接并管理的 Hypervisor（即 RHEVH）。RHEVH 具有两种实现方式：1、安装包含了 Hypervisor 的操作系统；2、安装专为 RHEVH 设计的操作系统。第二种方式安装等 RHEVH 能更高效的利用 H 端的硬件资源。

在 HOST 上安装 Hypervisor 软件，将 RHEL HOST 配置成为 RHEVH 端使部署 RHEV 环境变得更加简单，成功率更高。RHEVH 模拟了底层硬件来支持虚拟机的运行。以 Hypervisor 微型系统来实现的 H 端占用很少的物理资源就能支持虚拟机的正常运行，因为它只包含 RHEL 中支持虚拟机正常运行的所需代码的一部分内容。Hypervisor 最重要的功能是当 Hypervisor 接受到敏感指令时，首先判断

此指令是由虚拟机发出还是宿主机发出，如果判断时是由虚拟机发出，Hypervisor 会将此指令模拟翻译后交由宿主机 CPU 执行此指令。

2.5.3 LDAP

在 RHEV 环境中每个角色拥有不同权限，用户的权限可以依靠每个用户指定不同的角色来实现。LDAP/IPA(Linux)、AD(Window)为用户提供了认证体系。

LDAP（Lightweight Directory Access Protocol）轻量级目录访问协议是一种用于访问目录服务的应用协议。它对 X.500 目录访问协议继承的同事也对其进行优化和修改[43]。LDAP 也是一种标准的、中立的、可扩展开的、开放性的目录访问应用协议[44]。该协议采用 C/S 模式，是管理应用程序和目录读/写操作的运行在浏览器上的应用程序。LDAP 每个目录的条目（entry）由各种属性（attribute）的聚集而成[45]，每一个条目有唯一引用，称为专有名称，英文是 DN（Distinguished Name）。LDAP 目录呈树状结构组织，因为其查询、索引和搜索上高效率，被称为优化了的专业分布式数据库[46]。

其特点如下:

1、LDAP 的结构组织用树来表示，而不是用表格使其查询、索引速度很快。

2、LDAP 可以很快地查询结果，写入数据较慢。不支持事务处理、事务回滚等复杂操作，这也是它的组织形式决定的。

3、LDAP 提供了静态数据的快速查询方式。

4、LDAP 使用 C/S 模型，S 端存储数据，C 端操作目录。

5、LDAP 方便用户备份数据和恢复数据。

使用 LDAP 服务器的主要好处在于:

1、可以将 LDAP 作为整个组织的信息的集中管理点。由 LDAP 统一管理。

2、支持安全套接协议层（SSL）和传输层安全协议（TLS）等安全防护协议，LDAP 使用这些安全协议可以很好的保护重要信息，有效防止信息在网络上泄露。

3、LDAP 支持多种任务类型，如应用配置存储、系统信息存储、用户认证、用户权限认证、资源追踪、组织演示等多种任务类型。

4、LDAP 通常用于单点登录[47]，允许用户在多个服务上使用同一个密码。

2.5.4 REST API

REST 是 Representational State Transfer 缩写，中文为表述性状态转移。它是一个设计架构风格，由 Roy T. Fielding 在其博士论文中提出[27]。它专注于特定服务资源以及这些服务的表述形式。它代表了服务器上的一个特定管理项，资源的表述形式作为关键的信息抽象层。客户端执行标准的 HTTP 方法，如 GET、POST 方法等向服务器发送请求。客户端和服务器之间无状态的通讯，由客户端发出的每个请求必须包括为了完成此请求所需的所有信息，并且此请求其他请求相互独立、互不影响 。

REST 的设计约束有以下几项: 1、所有事物都被抽象为资源; 2、每个有唯一的资源标识符 URI; 3、对资源进行操作时使用统一的接口; 4、以资源的表示来处理资源; 5、消息都具有描述性; 6、所有的交互都是无状态的。

RHEV 采用 REST 的设计风格，API 进入点和 API 访问的资源集合的关系。

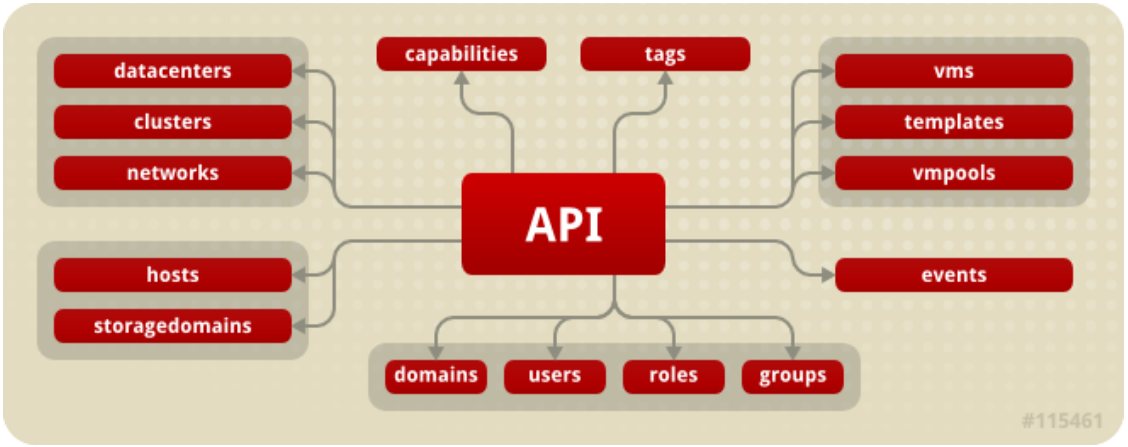

图 2.6 API 进入点和 API 访问的资源集合的关系

访问进入点会获得 API 可以使用的资源集合的 link（连接）项和 URI，每个集合都使用关系类型来指定客户端需要的 URI。link 项也为特定集合包括了一组 search URI，这些 URI 使用 URI 模板来集成搜索查询。URI 模板的目的是接受使用查询参数的自然 HTTP 特征的查询表述。客户端不需预先知道 URI 的结构，只需通过 URI 模板库来访问。每个搜索查询 URI 模板都使用关系类型来代表（格式是"collection/search"）。

表 2.1 搜索查询 URI 相关的关系

关系	描述
datacenters/search	查询数据库
clusters/search	查询主机集群
storagedomains/search	查询存储域
hosts/search	查询主机
VMs/search	查询虚拟机
disks/search	查询磁盘
templates/search	查询模板
VM Pools/search	查询虚拟机池
events/search	查询事件
users/search	查询用户

2.5.5 存储

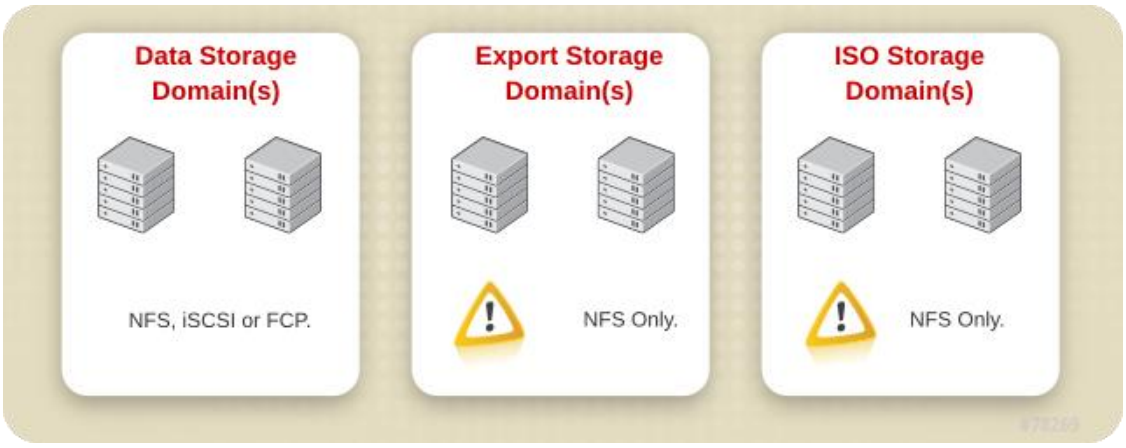

图 2.7 存储内部逻辑结构

在 RHEV 环境中，所有虚拟机的磁盘镜像、虚拟机环境信息、模板、快照和 ISO 文件等资源统一由中央存储系统进行保存。系统又将所有的存储分成不同的逻辑组，即是存储池。存储池和一组存储空间以及存储空间中的数据构成一个存储域。每个存储域根据存储的内容的不同划分为数据存储域、导出存储域和 ISO 存储域。

数据存储域存储数据中心中所有虚拟机上的虚拟镜像、虚拟机环境信息、模板和快照等信息。每个数据域只能被该数据域所属数据中心单独使用，不能与数据中心共享。数据域的数据类型由所属数据中心的数据类型决定，并且必须与数据中心数据类型一致。

导出存储域是一个用于存储临时数据的存储仓库。它是在数据中心和 RHEV 间复制和迁移数据镜像临时空间，来备份虚拟机和模板。虽然导出域只能被所属数据中心独自使用，但是 RHEV 允许它在不同的数据中心之间进行迁移。

ISO 存储域主要用来存储各种 ISO 文件，包括操作系统 ISO 和应用程序 ISO 文件。ISO 域可以被不同数据中心共享。

对与每个数据中心而言，必须包含一个数据存储域。是否包含导出存储域和 ISO 存储域由用户的需要决定。任何数据存储域只隶属于一个数据中心，数据存储域中的数据对所属数据中心中的所有主机是共享的。

存储网络可以通过使用 NFS（Network File System）、iSCSI（Internet Small Computer System Interface）、Gluster FS 或 FCP（Fibre Channel Protocol）实现。

2.5.6 网络

RHEV 环境中网络架构是针对 RHEV 环境中的不同对象之间的网络连接问题而抽象出来的。同时，它还被用来实现网络的隔离。其内部架构图。

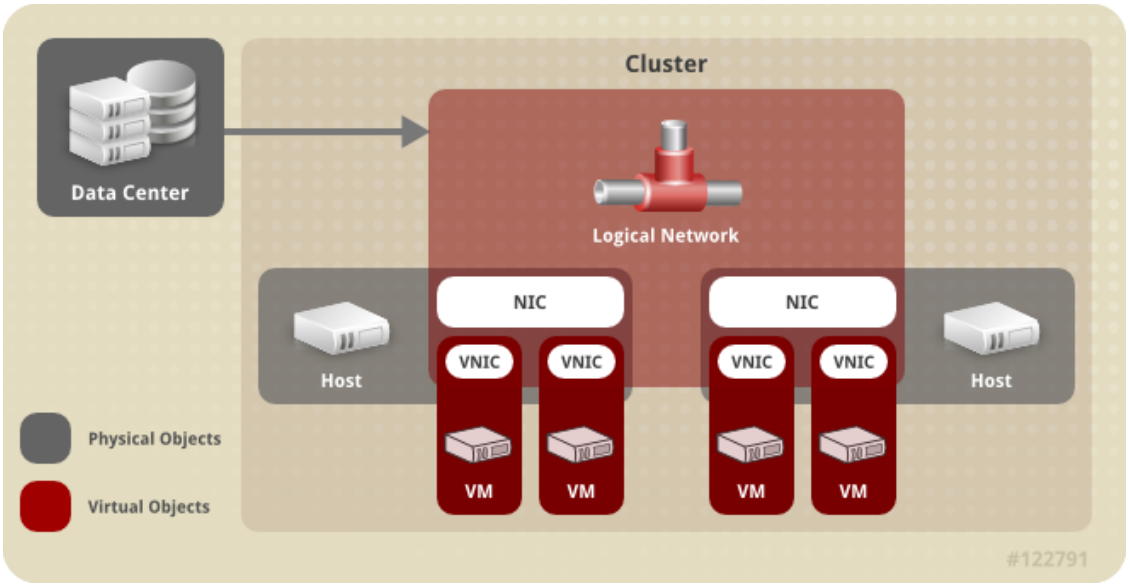

图 2.8 网络内部逻辑关系图

网络架构需要在 RHEV 的不同逻辑层上被定义。网络的实现需要底层物理网络架构的支持，同时保证物理网络架构必须支持硬件与 RHEV 中逻辑组件之间的网络连接。RHEV 网络架构需要一定等硬件设备和抽象设备的支持。所需设备如下：1、网络接口控制卡，也就是网卡，英文简称 NIC。它是一个将主机连接到网络中网络接口设备。2、虚拟网卡，英文简称 VNIC。它是利用 NIC 实现虚拟机之间的网络连接的抽象设备。3、绑定，即 bond。它是一个由多个网卡组成的网络接口。4、网桥，即 bridge。

逻辑网络是数据中心中网络资源的抽象。数据中心中的主机可以根据相关的参数修改分配给主机的逻辑网络参数。所有数据中心在为用户配置逻辑网络之前都默认名为 rheVM 逻辑网络来对网络进行管理。逻辑网络的目的是为 RHEVM 与主机的数据管理提供网络平台。在虚拟环境中逻辑网络在不同层次上被定义不同的意义。在数据中心概念层，默认由名为 rheVM 逻辑网络管理网络，同时用户可以根据自己的需求添加不重复的逻辑网络。新添加的逻辑网络需要被添加到网络所在的集群中才能进行网络管理。在集群概念层，默认每个集群都必须连接到相应的管理网络中。集群所属数据中心中的未被当前集群使用的逻辑网络可以被添加到当前集群。对于添加到集群的网络可以分为必须添加和可选添加，“必需的”网络被添加到集群中时必须被添加到所添加集群中的所有主机上，“可选的”网络对此不作要求，可根据用户需求添加到所属集群的主机上。在主机概念层中，对于虚拟机逻辑网络由逻辑网桥实现网络连接。它要求每个逻辑网桥必须连接一个特定的网络接口。对非虚拟机逻辑网络连接不作要求，直接和主机上的网卡相关

联即可。在虚拟机概念层中，虚拟机上逻辑网络连接方式与在物理机上网络连接的方式相同。虚拟机通过虚拟网络连接到所属主机上的逻辑网络的同时也可以使用虚拟网络中的资源。

2.5.7 数据中心

数据中心是 RHEV 环境中的最高级别的抽象概念，它看作是由三个子容器构成的数据容器。三个子容器分别是：1、存储数据中心中存储类型信息、存储域的信息以及存储域间的连接信息的存储容器。这是针对数据中心专门抽象出来的概念，它可被所属数据中心中的所有集群共享；2、存储数据中心中逻辑网络相关的信息的网络容器。它也是针对数据中心抽象出的概念，并可以在集群中使用；3、存储集群相关的信息的集群容器。集群是虚拟机迁移的逻辑域。

2.6 本章小结

本章节介绍了 RHEV 虚拟化平台以及在本自动化管理系统中涉及的部分技术，如虚拟化迁移技术、RPC 技术、REST API 架构设计技术等。之后又介绍了关于 RHEV 的架构和基本概念以及 RHEV 的虚拟资源。

第3章 自动化管理系统需求分析

本章是对用户需求的分析和总结，目的为了明确用户基本需求以及自动化管理系统需要解决的问题，需要实现的基本功能。

3.1 自动化管理系统整体需求分析

本文以 RHEV 作为服务端为系统提供复杂的虚拟化实现。本章节先分析企业在业务处理遇到的一些问题，根据这些问题和用户的基本需求分析总结出系统需要完成任务和必要的功能点，之后逐一对各功能点进行细分。设计目标及解决的问题的分析:

1 、避免可控风险的需要

业务处理过程中存在的数据丢失、数据不完整、用户操作不当或非法操作、用户权限不足等一系列导致业务无法正常进行的风险，造成效率低下、用户积极性不高的影响。本系统需要在用户输入很少数据信息的情况下能保证业务的正常运行和业务的完全自动化的执行。

2、保证执行环境和结果正确的需要

用户在执行复杂的业务时需要考虑此业务依赖其他业务是否正确完成，和执行环境的各项配置是否正确，所依赖的业务是否返回正确的结果等问题。这给用户带来很大困扰和对业务执行也是很大阻碍。本自动化管理系统需要自动屏蔽这些依赖关系，用户只需考虑此业务相关问题提交给系统自动化实现，

3、保证操作的一致性的需要

用户手动操作成百上千次同样操作时，几乎不能保证操作的不会出现错误。在每次启动一个虚拟机后需要执行较多预处理和后处理脚本，一些脚本不能很好的捕获错误和异常，从而操作的失败。而且每次脚步需要与其他软件进行交换时不能保证每次都能取到正确的信息。所以本系统需要保证操作的一致性并能有效的捕获处理异常。

4、自动执行当前任务的需要

本自动化管理系统需要保证在用户提交了申请以后，业务逻辑层创建的任务被系统自动调度和执行。包括依据已存在的模板自动创建虚拟机; 自动创建虚拟机池,根据用户选用的模板信息自动配置池的属性;在多个 RHEVM 的站点之间，自动迁移模板等。

自动化管理系统需为用户提供一个简洁、易操作的、完整统一的操作界面，用户使用该前端界面能够提交虚拟机管理、虚拟机池管理、模板管理以及模板迁移的请求。在用户进行上述操作时，系统会在后台生成相应任务，以供服务端进行行调度和执行。同时将虚拟机信息、日志信息、任务信息、异常信息等保存至数据库(包括关系型数据库和非关系型数据库)。根据用户的需求分析本自动化管

理系统整体用例图如下

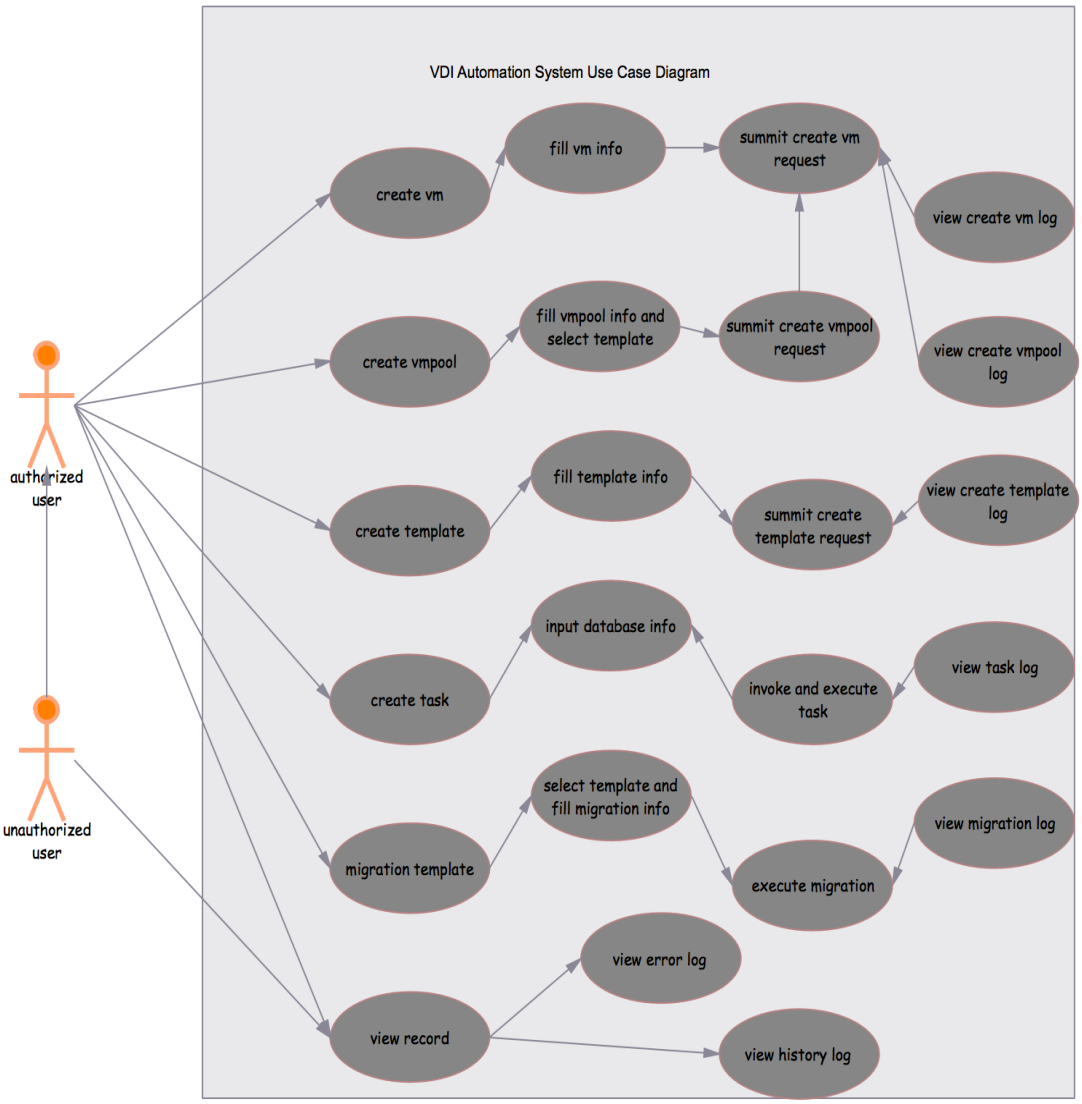

图 3.1 自动管理系统用例图

从上图我们更能清晰的看出本系统需要的一些基本功能。用户分成未授权用户和授权用户两种，授权用户权限最大可以对系统中所有模块都有操作权限，普通用户只能查看部分历史记录和异常记录。用户发起的虚拟机创建，虚拟机池创建，模板创建，模板迁移的请求，创建定时任务，都会被系统提交给后台生成相应的任务。后端任务调度层会根据任务设定的时间点去调用此任务并执行。在任务执行完成后，将执行过程生成的日志文件和执行结果返回到前端页面，展示给用户，同时进行数据持久化操作。根据任务处理的过程生成并保存各项日志的不同保存到不同的数据库中，以供用户以后查阅。

3.2 创建 VM Pool 的需求分析

虚拟机池中的虚拟机的权限在是在虚拟机池这一级来设定的，在 RHEVM 创建完成 VM Pool 时已经对其权限设置完成。用户在不同的时刻请求使用 VM Pool 时，RHEVM 会前后两次分配用户使用的虚拟机可能并不是同一台，而且每次用户使用完虚拟机，RHEVM 将会关闭该虚拟机，在需要启动时再重新启动，虚拟机在关闭后其中的数据会被清空。用户最好不要用虚拟机池中虚拟机来保存数据，但是可以选择中央存储设备中或可持久的保存新数据 IO 设备来保存数据。在创建完成虚拟机池之后，RHEVM 会自动的创建组成这个虚拟机池的虚拟机，并将其设置为处于“down”状态。当用户需要使用虚拟机时，虚拟机才会被设置成“up”状态表示虚拟机正在运行可以分配给用户使用。

由于虚拟机池中的虚拟机都是通过同一个模板创建的，池中的所有等虚拟机都共享一个只读磁盘镜像，共享同一网络资源，位于同一个集群中，但分配的 IP 不同。用户用完虚拟机之后系统都会将此虚拟机中数据删除，这意味着虚拟机池所使用的存储较小，只有模板相同的空间和用来临时存储用户使用数据的存储空间。从另一方面也说明了使用虚拟机池中的虚拟机比使用单独虚拟机要节省存储空间。所以在创建 VM Pool 时，用户只需要输入需要使用的模板，Pool 名称，Pool 大小，以及 Pool 需要所处的集群标识这些信息，在这些信息验证通过之后本系统会为此请求创建一个任务，由调用进程调度执行之后返回结果，生成记录。而中间的验证由 RHEVM 和本系统出来逻辑共同完成。

3.3 创建模板的需求分析

模板的创建依赖于用户选中源虚拟机以及源虚拟机上安装的软件和虚拟机环境变量信息。另外用户被推荐使用泛化（generalization）来创建虚拟机。泛化在 RHEVM 环境中是指在创建虚拟机式不考虑那些只与特定系统相关的、在不同的系统上具有不同值的信息。是否使用泛化来定制虚拟机的配置，对其不会有太大影响。Red Hat Enterprise Linux 虚拟机使用 sys-unconfig 进行泛化；Windows 虚拟机使用 sys-prep 进行泛化。如需了解更多关于在 Red Hat Enterprise Virtualization 环境中对 Windows 和 Linux 虚拟机进行泛化的信息，请参阅 Red Hat Enterprise Virtualization 管理指南。

当准备好一个源虚拟机之后，停止其运行，用户就可以基于该源虚拟机来创建模板。在创建模板的过程中，需要注意源虚拟机的磁盘镜像会被复制成一个只读的磁盘镜像文件。一旦模板创建完成，此文件无法进行修改，以后所有根据该模板创建的虚拟机都将使用此磁盘镜像文件配置信息。从这个角度来看，模板就是包含虚拟机环境配置信息的磁盘镜像文件。当然本自动化管理系统在创建模板时，会让用户选择是否按照源虚拟机的形式来进行创建或时用户手动配置 cluster 信息，cpu 信息，memory 信息等等必要信息进行创建。

3.4 模板的迁移需求分析

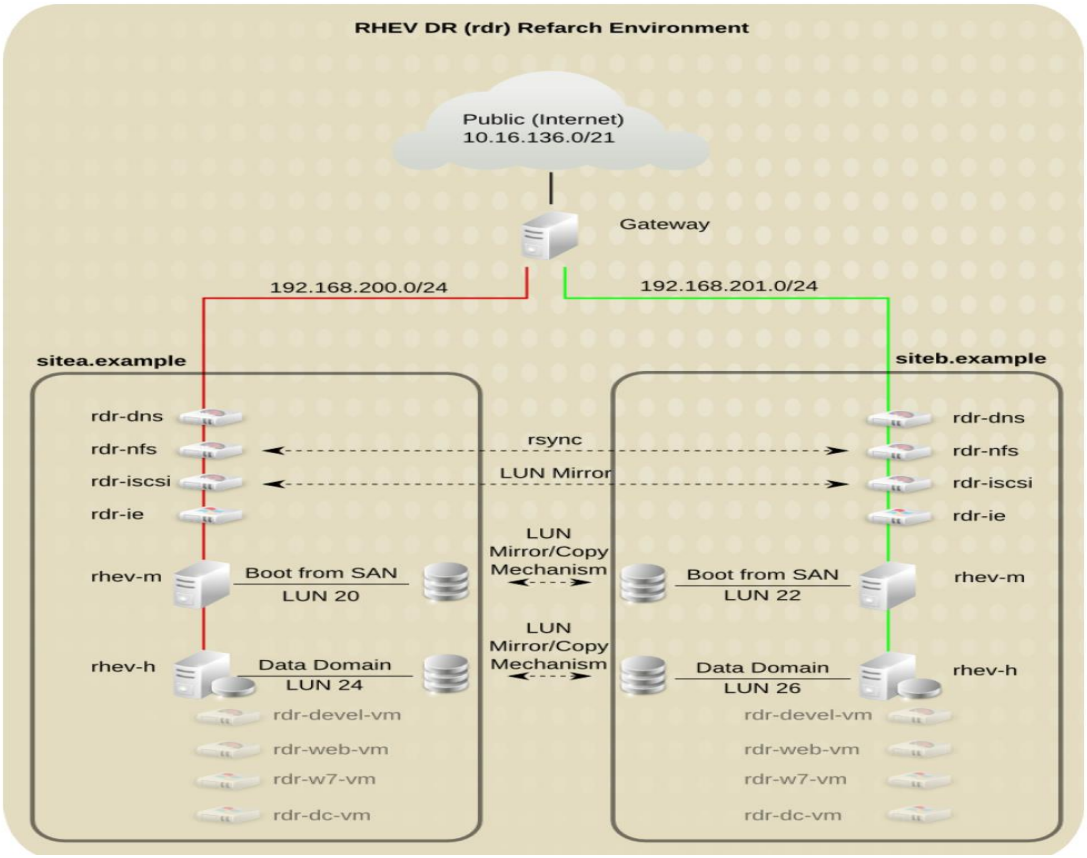

图 3.2 不同站点之间逻辑关系

在两个独立的站点之间故障迁移，对大型企业或事小型企业都是一件非常困难的事情。RHEV 在这方面给出的方案是依靠完备的前期预防措施和充足的数据备份，在最短的时间内让 RHEV 在指定的站点重新启动运行。其前期预防是数据备份过程再次不在阐述。由于每个站点是具有不同网络的单独网络地址和域名。每个站点之间的网络服务和 RHEVM 都不相同。两个不同站点通过网关系统连接在一起并连接到公共网络。其架构图大致如下。模板的迁移需要涉及存储域的迁移。存储域就是一组有一个公共存储接口的数据镜像，它包括了模板、虚拟机的数据镜像或 ISO 文件以及存储域本身的元数据。一个存储域可以由块设备（SAN--iSCSI 或 FCP）组成，也可以由文件系统（NAS--NFS、Gluster FS，或其它 POSIX 兼容的文件系统）组成。在 NFS 中，所有的磁盘、模板和快照都是文件。在 SAN（iSCSI/FCP）中，每个磁盘、模板和快照都是一个逻辑卷。逻辑卷由不同的逻辑块设备组成逻辑卷组根据 LVM（Logical Volume Manager）划分而成。逻辑卷作为虚拟硬盘供用户使用。逻辑硬盘可以有两种格式：QCOW2 或 RAW，存储类型可以是 Sparse 或 Pre-allocated。快照的类型是 sparse，但它可以是为 RAW 或 sparse 磁盘创建的。在不同站点之间 RHEV 是不能直接进行迁移的，所以两个站点之间

模板迁移不仅考虑存储域中文件格式问题，也需要考虑到站点之间连接的问题

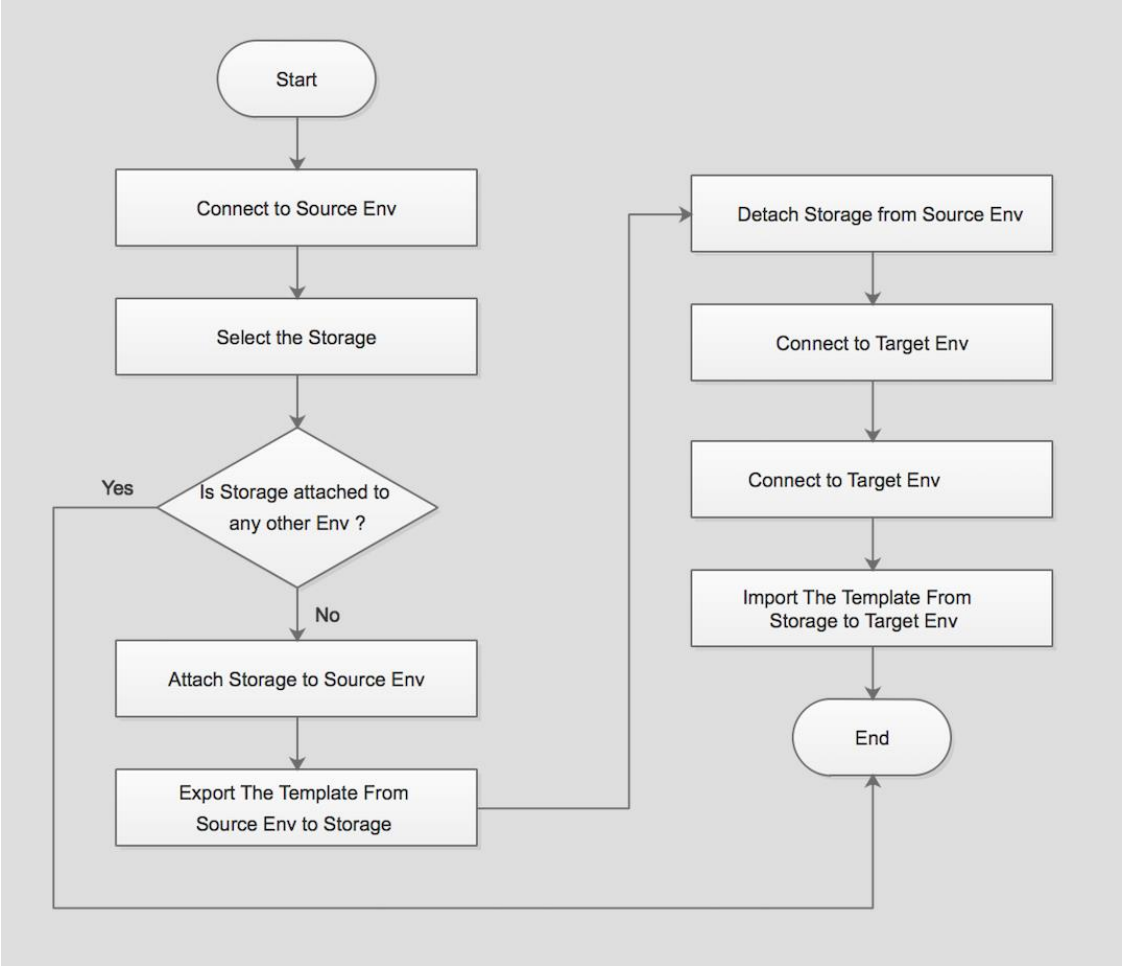

图 3.3 模板迁移的流程图

模板的迁移与 VM Pool、VM 的迁移方式不同。用户在选择要迁移的模板之后，需要用户给出需要迁移到站点位置，同时需要指定存储域的位置，如果需要迁移到的存储域的位置在指定的站点中也有挂载，则直接为此请求建立任务记录，并返回结果。这种情况相对简单。如果需要迁移到的存储域的位置不在指定的站点中挂载，过程会相对的复杂，迁移也会较为麻烦，需要先将存储域从原有站点卸载，连接到目标站点之后将此存储域挂载到目标站点，之后将所选站点的模板所有信息传入到目标站点中，并在此站点中创建新的模板。

3.5 创建 VM 的需求分析

创建虚拟机是指通过自动化管理系统创建获取主机上提供的虚拟的 CPU、内存、存储、网络、权限、集群、IO、GPU 等可用资源的过程，并在虚拟化服务器上生成一个可给用户提供部署应用服务的进程服务的请求，系统将此请求经过验证提交给 RHEVM，由其执行的完成后返回结果的过程。对于用户来说这只是发

送一个在一台服务器上添加一个或多个操作系统，并且按照不同的需求在其上部署应用程序的请求。

在自动化管理系统创建一个虚拟机需要用户通过管理界面输入虚拟机的名称，名称唯一，以使用的名字不能再用，这个名称可以作为虚拟机的索引来查询虚拟机的各项信息，也可以作为引用在控制面板中的显示进行显示。自动化管理系统根据用户提交请求时选用的模板的配置信息，在通过内部功能模块验证是否能够获取虚拟机配置信息，判断该虚拟机所使用的硬盘是否被别的虚拟机占用以及虚拟硬盘状态是否正常，虚拟网络是否能够正确分配 IP 地址，是否能够加入到相应的域，是否能够正常分配相应的权限，是否能否获取虚拟机正常地运行需要适当数量的硬件资源（比如 CPU、内存、硬盘空间、IO 等）等一系列的验证，之后将任务提交给 RHEVM 进行执行并返回结果。以上所说的资源都在创建虚拟机时进行分配。对于 CPU、内存、硬盘空间最小值的限制一般根据需要部署应用服务以及需要的操作系统的要求来定，不同的操作系统以及不同的应用服务需求的资源的最小值都不同，这就要求用户在创建虚拟机时做好可扩展性的准备。否则或造成创建虚拟机时能够支持当前应用，随着应用的增加和服务增多虚拟机并不能完全满足之后应用服务所需要的资源数量。面对这种情况还有一种方法来加以解决。在用户选择当的虚拟资源之后，RHEVM 创建一个可供调度进程调度的任务，由调度进程决定何时执行此任务并返回结果和执行记录。

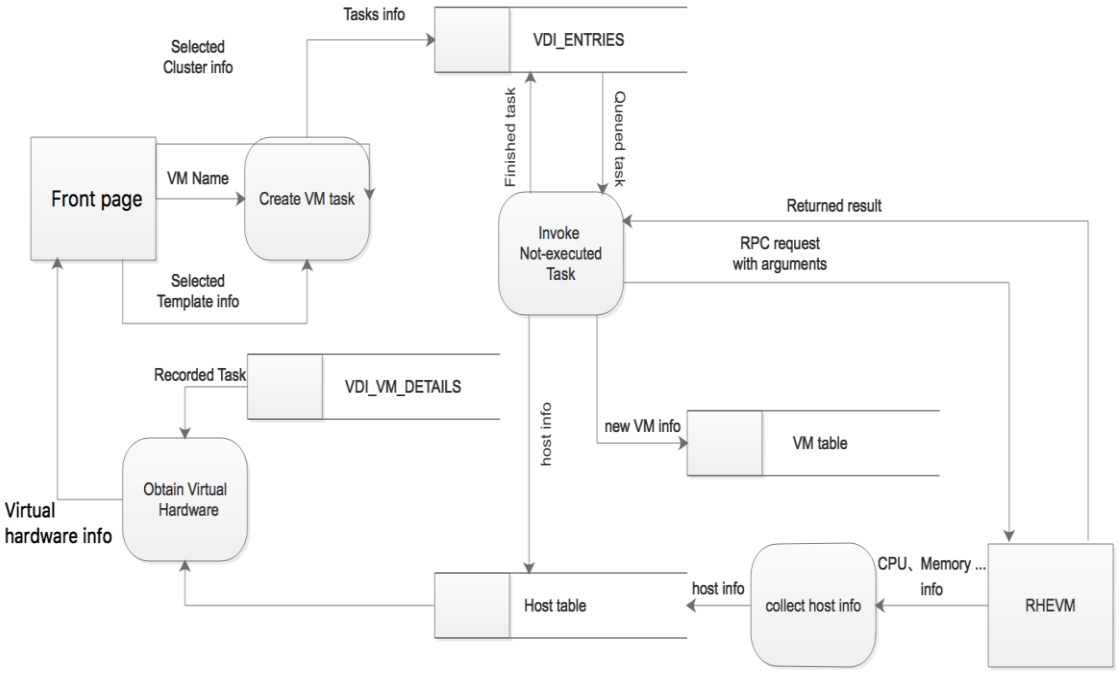

图 3.4 创建 VM 的数据流程图

如上图所示的数据流图中将用户选择的资源数目以及虚拟机名称创建状态为 submited 的请求，通过 RPC 将该请求传递到业务逻辑处理层，生成任务，并写入 VDI_ENTRIES 表中，之后由调度进程将任务调度到后端任务处执行逻辑单

元，任务执行逻辑调用 RHEV 提供等 REST API 创建 VM，在任务执行完成之后。调度进程将执行结果返回前端页面，同时写入数据库中。

3.6 本章小结

本章主要介绍了自动化管理系统的需求，根据分析本管理系统需要实现 VM 的创建、VM Pool 的创建、模板的创建、模板在不同站点之间的迁移等功能。

第4章 自动化管理系统的整体设计与技术实现

在上一章中介绍了自动化管理系统设计目标及解决的问题，对创建 VM Pool、修改 VM Pool、创建模板、两站点之间模板的迁移、创建 VM、修改 VM 信息和创建定时任务等需求的内部逻辑进行分析。明确了用户的基本需求和本管理系统需要实现的基本功能点。本章在上一章分析的基础之上对本管理系统进行系统架构设计，数据库设计，各功能逻辑实现设计。本自动化管理平台是基于已经搭建完成并能正常运行的 RHEV 环境进行进一步开发的项目。由于此项目参考本人在实习公司的参与开发的实习项目，实习项目的运行依赖实习公司 Cloud 环境以及公司其他项目正常运行。本文默认的系统环境是多个站点能够同时运行 RHEV 环境，并支持本管理系统正常运行。

4.1 自动化管理系统整体实现

4.1.1 整体架构设计

本管理系统基于 B/S 架构来实现。整个管理系统分为：UI 模块属于前端页面层，通过浏览器对管理系统进行操作或展示用户请求的信息，此部分使用 Js+JQuery+HTML5 技术进行实现；内部功能模块属于业务逻辑控制层，主要是实现业务处理、数据转换和数据存取，此部分利用 Java 技术进行实现；数据库功能模块既涉及业务逻辑处理层，也涉及后端任务调度层，使用开源型数据库 MySQL 来存储关系型数据，Mango DB 来存储非关系型数据（保存 VM、VM Pool、模板信息时使用，用户通过发送请求进行查看）。对于前端 UI 与后端的数据交互可以通过 AJAX 进行实现。

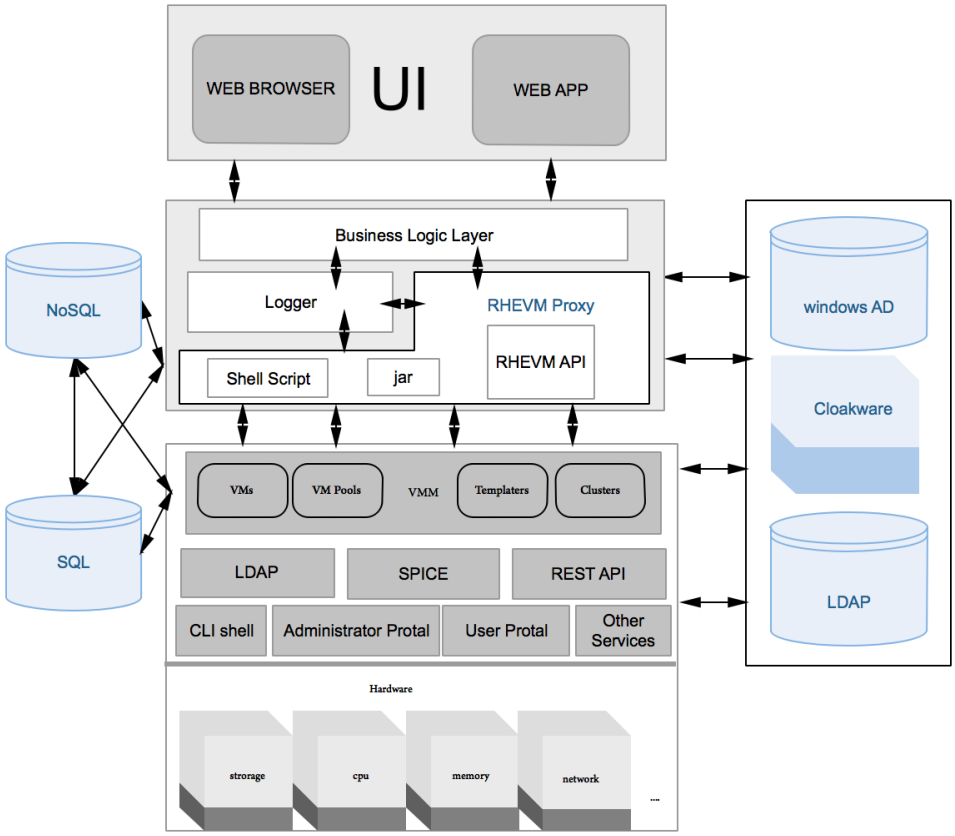

图 4.1 自动化管理系统整体架构图

用户登录时验证用户是否有本管理系统的使用权限，无权限者提示权限不足，用户需要被授权之后才能成功登录。用户成功登录以后，在前端页面会显示当前 RHEV 中所有的 VMs、Hosts、Users 的统计结果——即当前系统中正常运行的 VMs 和未正常运行的 VMs、正在被使用的 Hosts 和未被使用的 Hosts、正在使用 VM 工作的用户和未使用 VM 工作的用户的数据展示。在 Create 功能模块区包含 create VM task、create VM Pool task、create Template task、create 迁移 task 四个功能点。在 View 功能模块区包含 view VM、view VM Pool、view Template、view 迁移等功能点。

用户通过前端页面点击相应功能图标分别进入不同的功能模块，自动化管理系统根据用户使用的功能不同，让用户输入不同的信息，如模板名、VM name、VM Pool name、CPU 数目、内存大小等、执行任务的时间点。本系统会将这必要的用户输入信息以参数的形式写入用户发出的 RPC 请求。

后台功能模块在接收到 RPC 请求中的参数之后，向服务器发送请求得到一些必要的信息，如 VM ID、permission ID、group ID 等信息。功能模块将这些信息

和用户提供的信息加上时间戳后包装成 process.controller，并将之写入数据库，同时后台调用 Job Quartz 判断是否需要立即执行任务。

如果任务被判断为需要立即执行，Job Quartz 会为此任务创建线程，并执行任务返回任务结果和生成任务执行日志。否则任务将被保存在数据库中等到下一个轮询周期判断是否需要执行此任务。在任务执行完成之后服务器会把返回结果写入到对应的数据库中，同时将日志文件文件写到非关系型数据库以备查看。

4.1.2 数据库设计

本系统将使用数据库产品主要以开源产品为主,包括: MySQL 和 Mango DB。使用他们的原因: 1、两款产品都是开源产品，无需支付使用费用。2、MySQL 和 Mango DB 都是目前开源社区使用最多产品之一，支持社区较多。3、两款产品的高性能、高可用、易扩展等优良特性很好的满足了系统对数据库等要求。根据上一章的设计，本系统需要有 VM 表、VM Pool 表、permission 表、template 表、host 表、datacenter 表、disk 表、cluster 表、network 表、groups 表、role 表、domain 根

据 RHEV 虚拟化设计，这些是实体其内部逻辑。

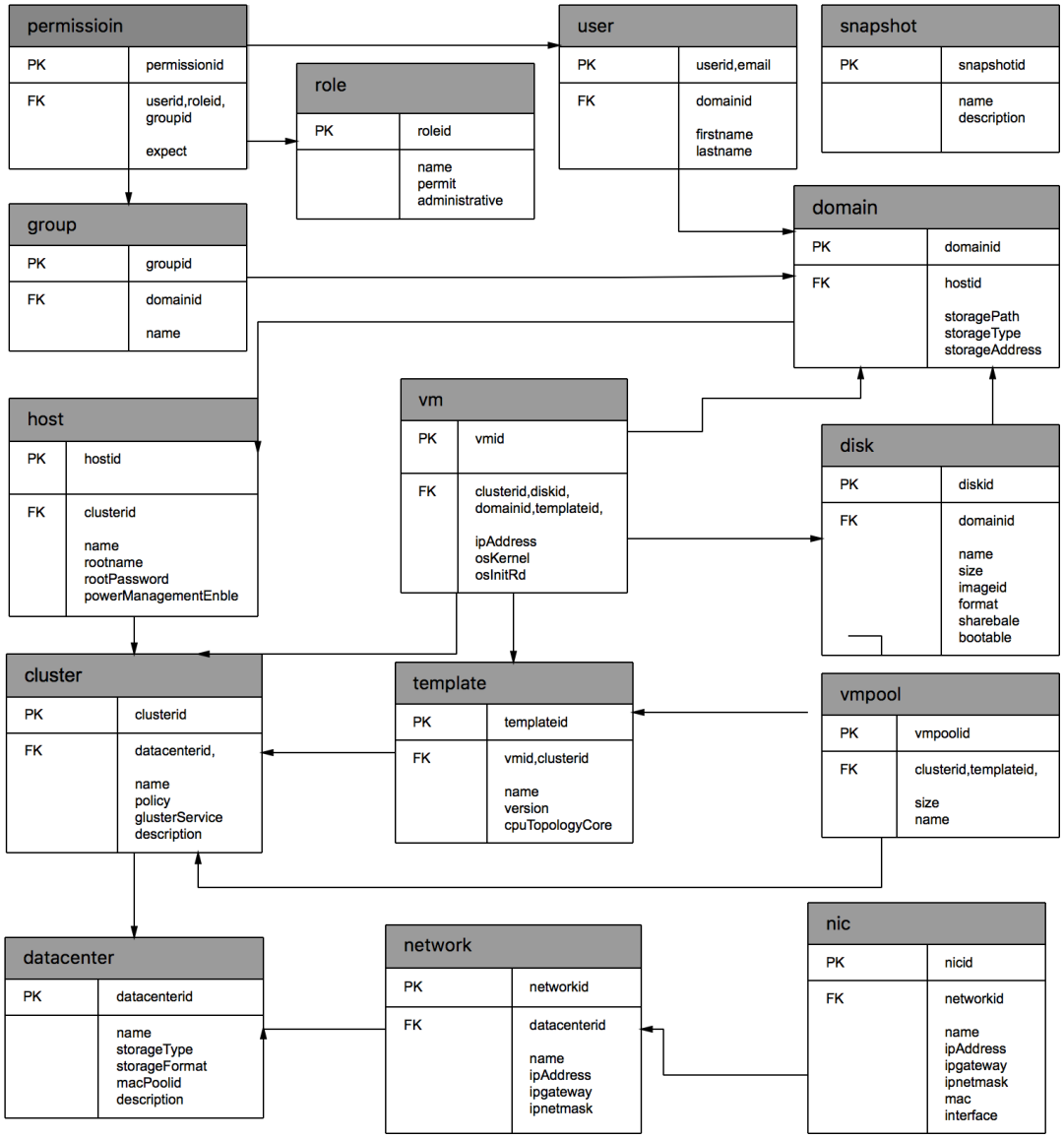

图 4.2 各实体之间联系

图中关系每个 VM 可以附加多个磁盘，每个附加磁盘只能为一个 VM 提供服务。每个 VM 只能隶属于一个主机中，每个主机容纳多个 VM。一个主机隶属于一个集群，一个集群包含多个主机。每个 VM 都是根据一个模板进行创建的，一个模板可以创建多个 VM。VM Pool 中只能有个一个模板，每个模板可以属于不同的 VM Pool。一个 role 可以有多个权限，每个权限可以分配给多个 role。一个组里可以有多个用户。一个用户可以在多个域中，每个域中也有多个用户。每个 host 有多个域，每个域也包含多个 host。每个集群和逻辑网络只能在一个 datacenter 中。

以外健连接来表示各表之间的逻辑关系。以上中的表是 RHEV 中基本的表，它们分别代表 RHEV 中的不同的实例，同时又代表 RHEV 中不同的资源。

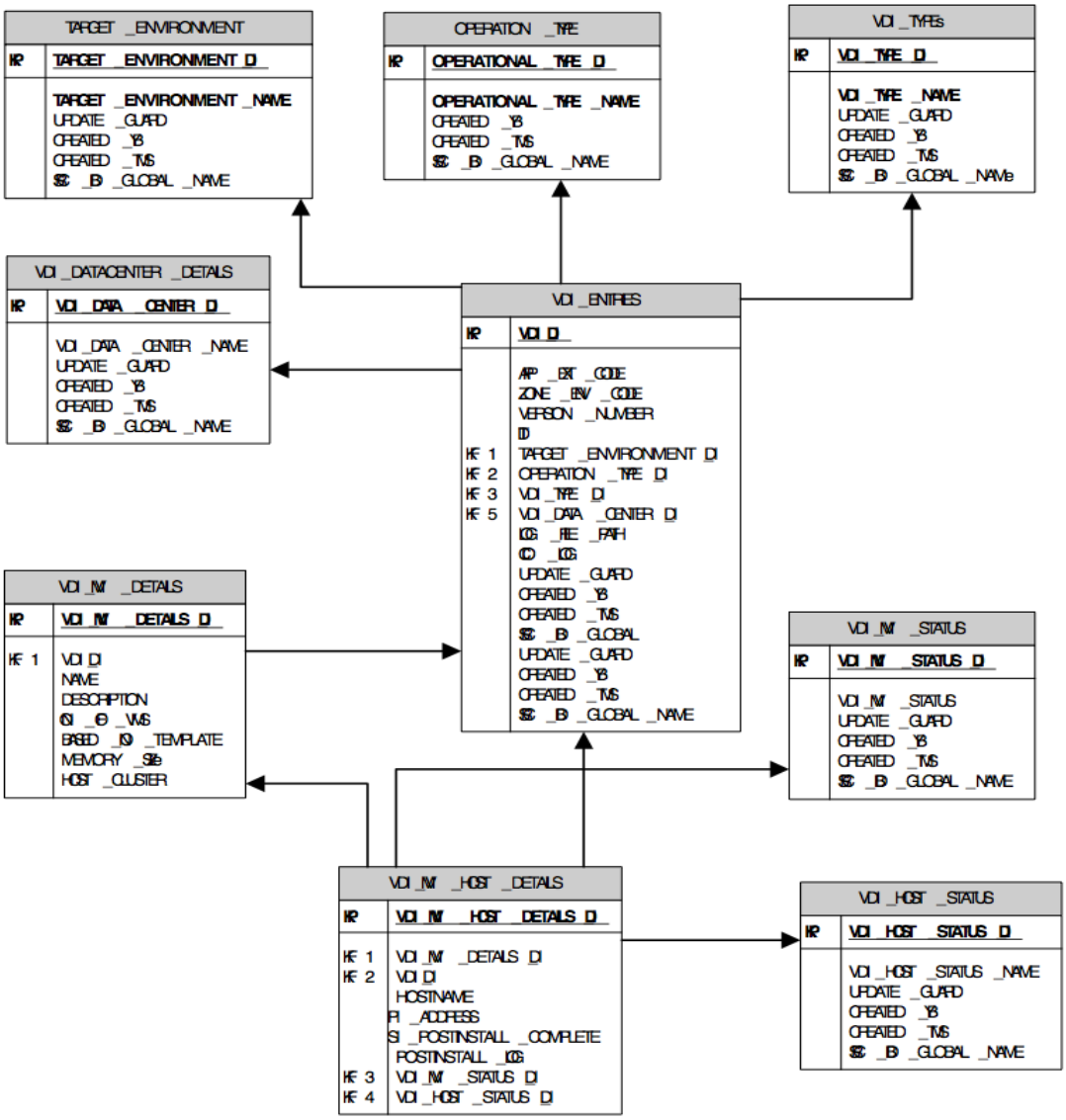

图 4.3 各任务之间的联系

根据本自动化管理系统中不同业务逻辑，用户的每次请求，本自动化管理系统都将为其建立一个任务，以下图中不同的任务表代表不同类型的任务，其内部逻辑

VDI_ENTRIES 记录执行将要执行的任务。调度进程将选择其中的记录传回后端 Job Quartz 进行执行。

VDI_DARACENTER_DETAILS 表记录在每次任务执行后等数据中心的变化，并记录当前数据中心的数量和状态等信息。

VDI_VM_DETAILS 记录创建 VM 的任务或修改 VM 信息的任务，任务执行之后，Job Quartz 将根据该任务记录的主键（即字段 TID）将执行结果写入该表，

修改任务的状态值的同时，将执行任务过程中生成的日志文件保存到相应的数据库中。。

VDI_TEMPLATE_DETAILS 记录所有创建模板的任务，在所创建的任务执行完成之后，Job Quartz 将根据该任务记录的主键（即字段 TID）将执行结果写入该表，修改任务的状态值的同时，将执行任务过程中生成的日志文件保存到相应的数据库中。。

VDI_VM POOL_DETAILS 记录创建 VM Pool 或修改 VM Pool 的任务，每次任务执行完成之后，Job Quartz 将根据该任务记录的主键（即字段 TID）将执行结果写入该表，修改任务的状态值的同时，将执行任务过程中生成的日志文件保存到相应的数据库中。。

VM 存储当前 RHEV 中所有 VM 的信息。每个 vmid 和 name 唯一表示一个 VM。其中 VM 的配置信息也将存储在此表中。templateid 和 clusterid 分别表示 VM 使用的模板和 cluster。由于真实的 VM 中要涉及图形协议、视频类型、外设使用策略、显示器数量等等较多配置，本论文中做了取舍，只在 VM 表中设置了较为重要的字段，其他未涉及字段因为与本论文业务关系不大并为在本论文体现

表 4.1 VM 表结构

字段名称	数据类型	是否允许非空	字段描述	规则
VMid	varchar(100)			
name	varchar(30)	N	虚拟机名称	主键
clusterid	varchar(100)		cluster 主键	外健
templateid	varchar(100)		模板表主键	外健
omainid	varchar(100)		domian 表主键	外健
diskid	varchar(100)		disk 表主键	外健
ipAddress	varchar(30)		虚拟机 ip 地址	
osKernel	varchar(30)		操作系统内核	
osType	varchar(30)		操作系统类型	
oscmdline	varchar(30)		命令行模式	
fqdn	varchar(30)		完全合格域名	
memory	double		占用内存大小	
cpuTopologyCore	int		分配 cpu 核心数	
decription	varchar(200)		VM 等描述	

表 4.2 VM Pool 表结构

字段名称	数据类型	是否允许非空	字段描述	规则
VM Poolid	varchar(100)		虚拟机 ID	
name	varchar(30)	N	虚拟池名称	主键
clusterid	varchar(100)		cluster 主键	外健
templateid	varchar(100)		模板表主键	外健
runningVMs	int		正在运行的虚拟机的数量	
size	double		池中虚拟机数量	
assignedVMs	int		已分配给用户的虚拟机数量	
decription	varchar(200)		VM 等描述	

表 4.3 模板表结构

字段名称	数据类型	是否允许非空	字段描述	规则
templateid	varchar(100)		模板 ID	
name	varchar(30)	N	虚拟池名称	主键
clusterid	varchar(100)		cluster 主键	外健
VMid	varchar(100)		源虚拟机的标识	外健
createDate	datetime		模板创建时间	
memory	double		模板占用内存大小	
status	varchar(30)		模板的状态	
cpuTopologyCore	int		模板分配到 CPU 核心数	
comment	varchar(200)		模板备注信息	
decription	varchar(200)		VM 等描述	

4.2 关键技术设计

4.2.1 创建 VM Pool 的设计

本自动化管理系统根据用户输入的 VM Pool name，template name，released number，effective time, pool description, pool size(<239)，batch size(<25)，memory size 等信息。系统先会根据 template name 向 RHEV server 请求整个模板的信息，从中解析出模板 d id、cluster id、domain id、vlan、storage 等必要的信息，并将这些字段和用户输入的部分信息写入 process.controller 中，并将 process.controller 写入 VDI_VM POOL_DETAILS 表中同时设置表中 deploystatus 为 0。调度进程在取得数据权限之后从数据库读取到此任务，判断此任务是否需要执行，如若不需要或有异常，将结果返回到前端页面，生成日志文件，调度进程继续调度其它任务。如若任务通过判断需要执行，调度进程将 process.controller 传给 Job Quartz job，由内部 RHEVM Proxy 模块解析出 VM Pool 基本信息和需要做的操作，通过调用 RHEVM REST API 创建 VM Pool。VM Pool 创建完成之后，Job Quartz job 会根据 VM Poolsize 和 batch size，逐一验证 VM 的状态是否启动、ip 地址是否分配。当确定所有的 VM 启动完成并配有正常的 ip 地址之后设置 deployStatus 为 1。同时生成 deploy.log， error.log，clo.log 等日志信息。将日志和执行过程生成的文件存储到数控库。完成以上步骤之后将 VM Pool 的 status 信息展现在页面中。下图是 VM Pool 的创建过程和 log 生成过程。

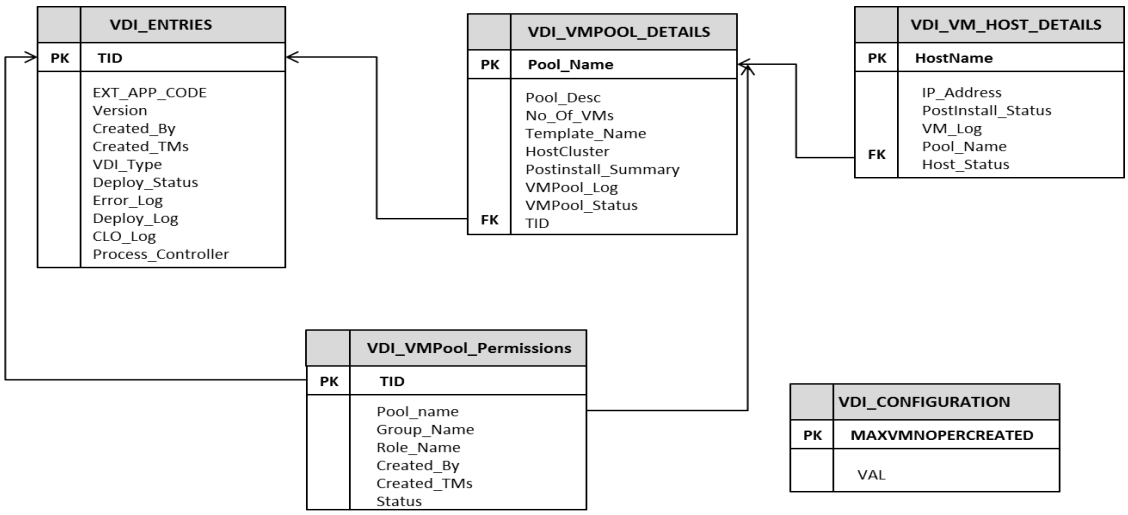

图 4.4 VM Pool 创建任务联系图

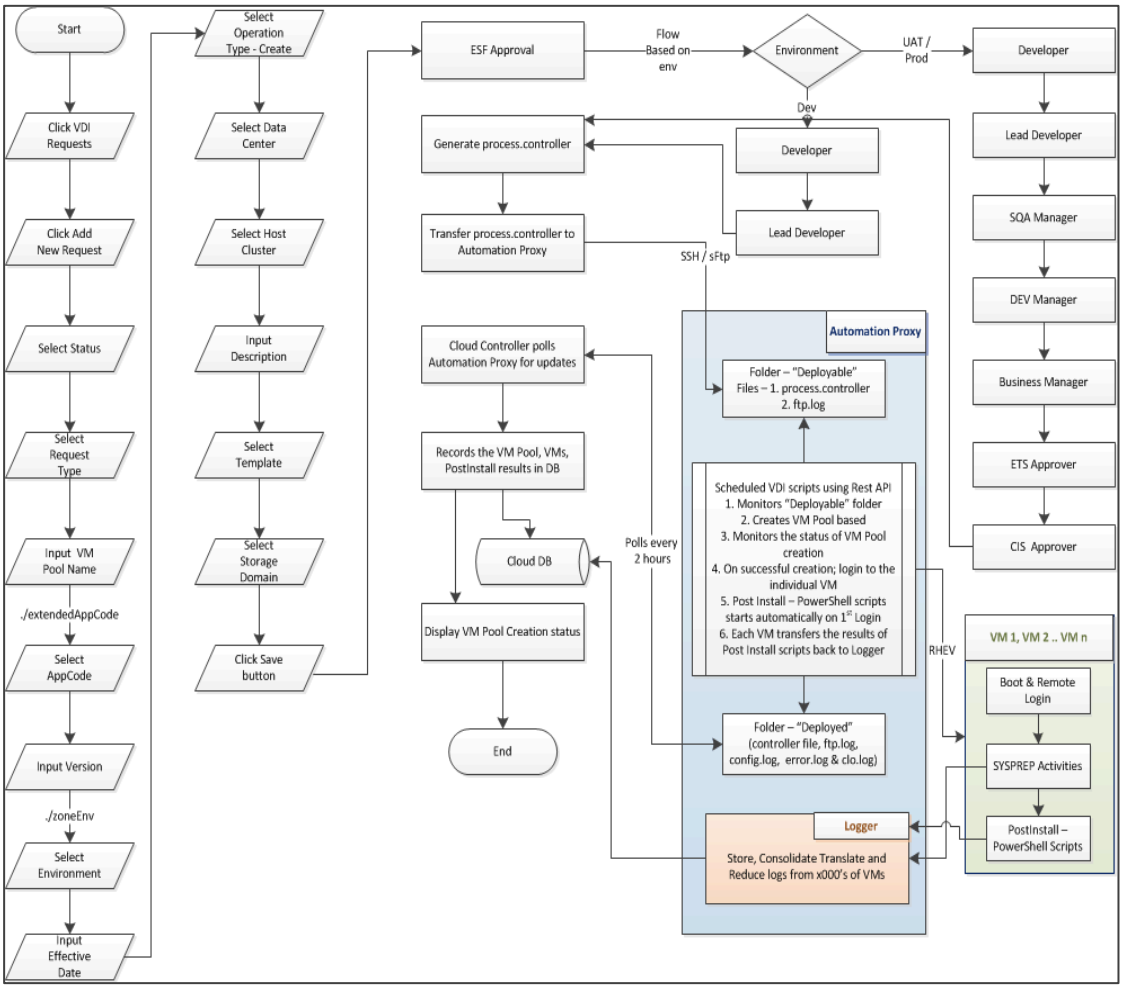

图 4.5 VM Pool 创建流程图

4.2.2 修改 VM Pool 的设计

在编辑 VM Pool 的时候，前端往后段传入的有 VM Pool ID 和用户修改的信息。自动化管理系统将会实现两种对 VM Pool 的编辑——修改当前 VM Pool 和删除当前安 VM Pool。就修改当前 VM Pool 的操作而言。用户可以为当前 VM Pool 添加 action 和 RVD role 或是修改 pool size 的大小。在用户选择需要的功能之后填写必要的信息。提交给后端进程，进程根据用户的请求创建相应的任务，并将用户输入信息和系统请求获得必要的信息写入对应任务表 process.controller 字段。添加当前时间戳后写入数据库表中同时设置表中 deploystatus 为 0，等待调度进程调度。调度进程在取得数据库权限之后从数据库读取到此任务，判断此任务是否需要执行，如若不需要或有异常，将结果返回到前端页面，生成日志文件，调度进程继续调度其它任务。如若任务尚未得到执行且未有异常，调度进程将从任务信息中解析出 process.controller 信息并将之传给 Job Quartz job，再由其内部 RHEVM Proxy 模块解析出 VM Pool 基本信息和需要做的操作，通过调用 RHEVM REST API 对 VM Pool 进行修改。VM Pool 修改完成之后，Job Quartz job 会根据 pool size 逐一验证 VM 的状态是否启动、IP 地址是否分配。当确定所有的 VM 启

动完成并得到能够正常使用 IP 地址之后设置 deploystatus 为 1。同时生成 deploy.log，error.log，clo.log 等日志信息。将日志和执行过程生成的文件存储到数控库。完成以上步骤之后将 VM Pool 的 status 信息展现在页面中。

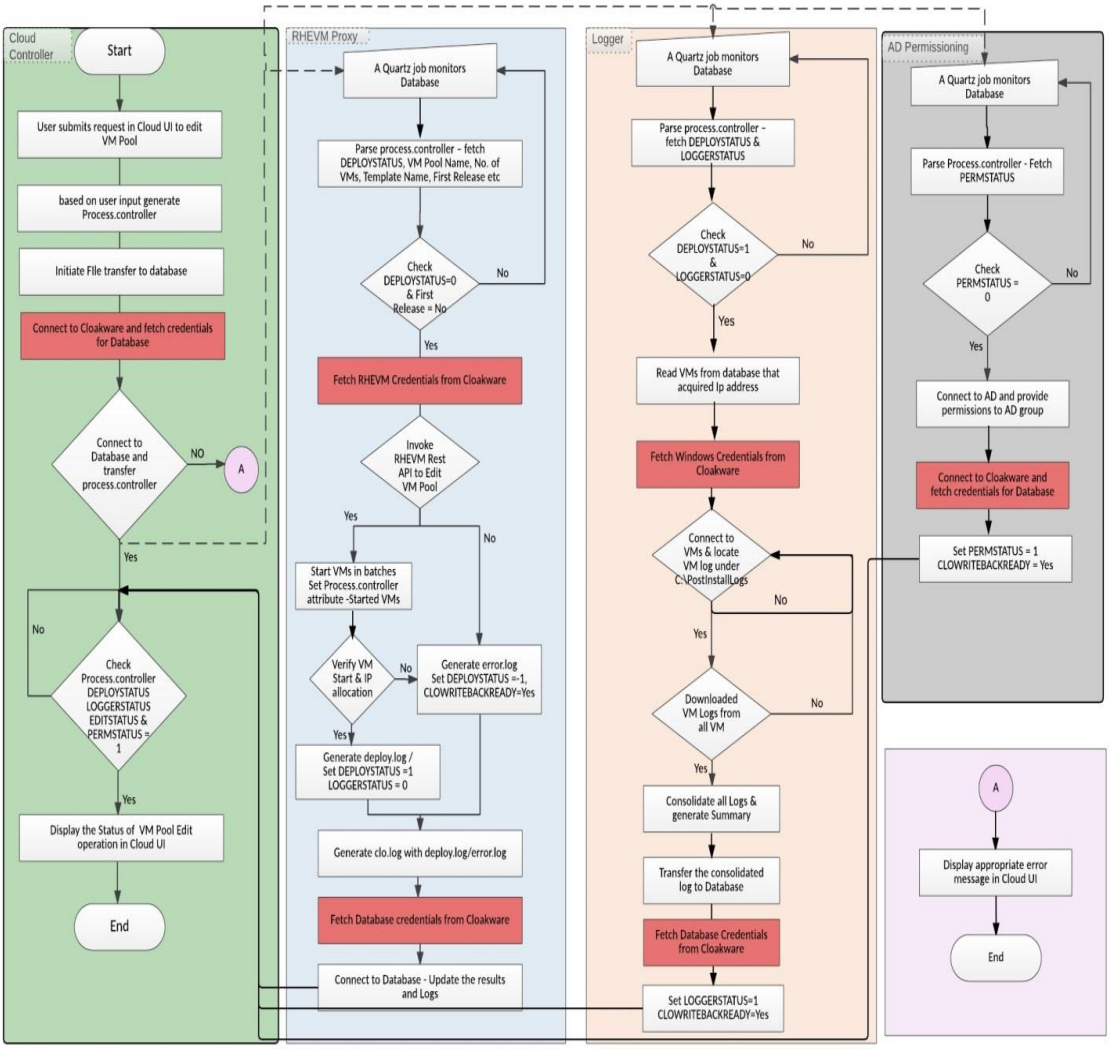

图 4.6 编辑 VM Pool 流程图

4.2.3 模板在两个站点之间的迁移设计

模板在两个站点之间的迁移，用户在前端页面指定，源站点和模板名，指定要迁移的目标站点 。自动化管理系统根据用户选择的源站点和模板名，向 RHEV server 请求获得 templateID 和模板中的 clusterID、源 VMid、模板占用的 memory 以及使用的 cpu 等信息，用户将模板迁移请求提交给后端进程，进程根据用户的请求创建相应的任务，并将用户输入信息和系统请求获得必要的信息写入对应任务表 process.controller 字段。添加当前时间戳后写入数据库表中同时设置表中 deploystatus 为 0，等待调度进程调度。调度进程在取得数据权限之后从数据库读取到此任务，判断此任务是否需要执行，如若不需要或有异常，将结果返回到前

端页面，生成日志文件，结束。如若任务通过判断需要执行，调度进程将 process.controller 传给 Job Quartz job，由它解析出模板基本信息、目标站点等信息以及需要做的操作，通过调用 RHEVM REST API 找到源站点和目标站点共享的 storage。由源站点获得 storage 的控制权，将需要迁移的模板从源站点复制到共享 的 storage 中，源站点释放控制权，再由目标站点获得 storage 的控制权，从 storage 中复制出模板到目标站点指定的 datacenter 中，释放 storage 的控制权。调用迁移 脚本更改必要配置即完成模板在两个站点之间的迁移。

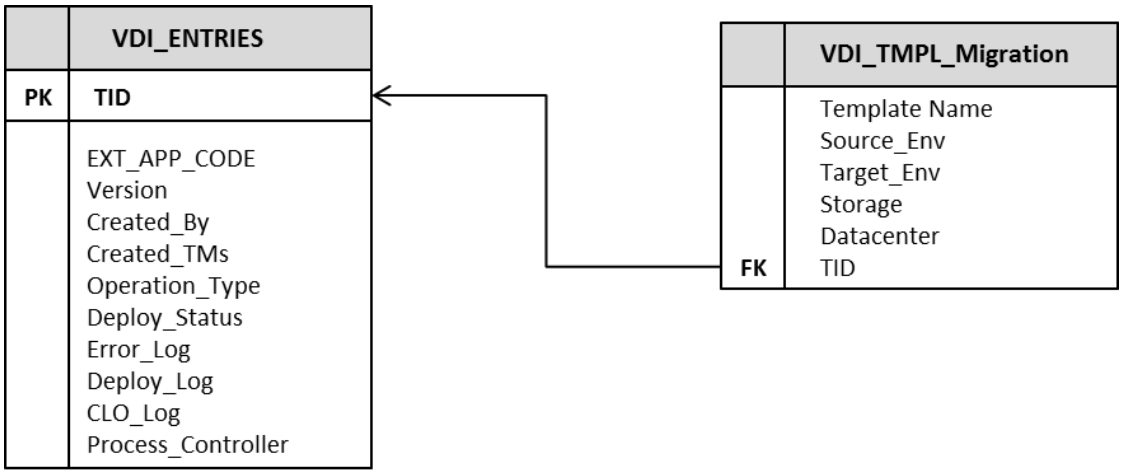

图 4.7 模板迁移任务之间的联系

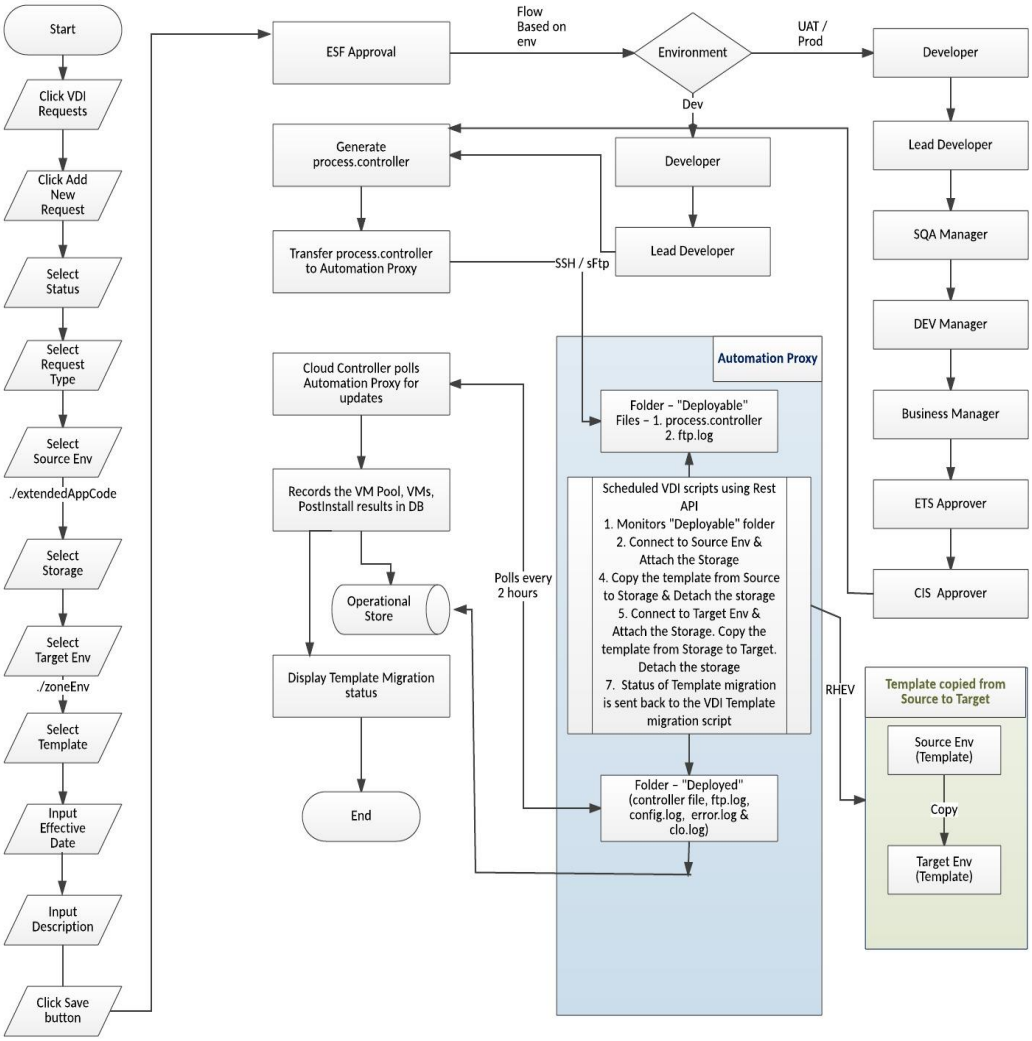

图 4.8 模板迁移流程图

4.2.4 调度进程和记录生成的实现

根据本系统的设计几乎所有的功能的实现都需要任务调度模块（即 RHEVM Proxy）的参与，所以任务调度模块在本自动化管理系统中也是最为核心的功能模块。同时在任务调度和执行的同时也伴随着记录日志的生成。在用户的请求提交之后，自动化管理系统验证用户身份信息是否合法，验证用户是否有权限发出请求，同时验证用户输入信息是否合法。在验证通过之后，自动化管理系统通过 RPC 将有用信息传给 RHEVM Proxy，由 proxy 对用户的请求进行进一步解析和执行。proxy 进程会通过访问 VDI_ENTRIES 表取得 process.controller 字段。根据解析的 process.controller 字段中的 action 参数的值，proxy 调用相应的 RHEV REST API，

同时将 process.controller 中的参数信息传入到 API 中。最终 RHEV 执行用户的请求，并将结果返回给 proxy。Proxy 会将返回结果以日志文件形式保存在 VDI_ENTRIES 表中。在任务被调入 proxy 的同时 Automation Proxy 模块中 Logger 模块也会读取 VDI_ENTRIES 中 deploystatus 和 loggerStatus 字段的值。如果发现 deploystatus=1 同时 loggerstatus=0（即任务已经执行，并未生成日志文件），logger 进程会继续查询 VM 以及 VM log 是否在制定的位置。如果上述条件得到满足，logger 进程会将所有得 VM log 合并，并生成日志概要，之后将 log 文件返回并写入数据中。

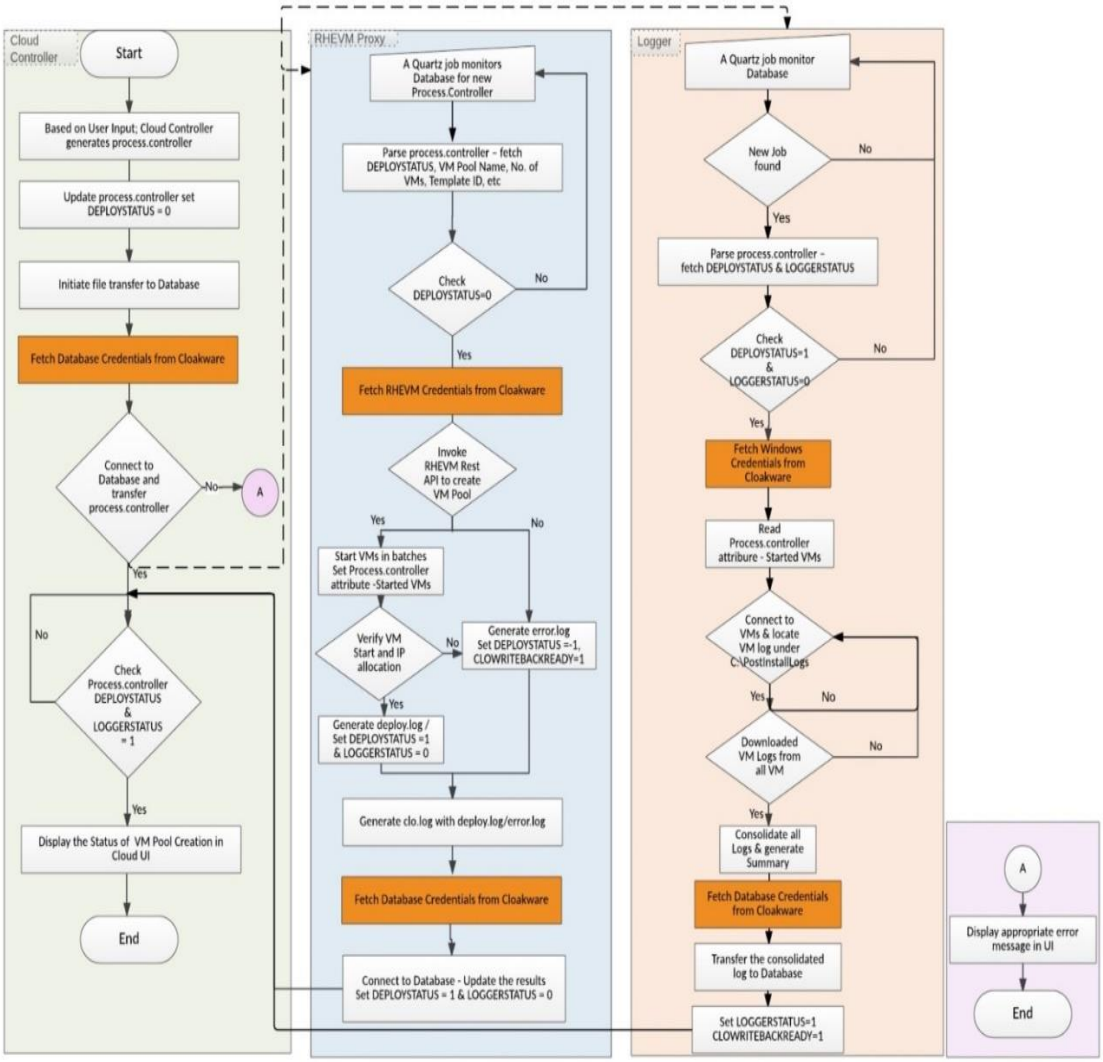

图 4.9 任务调度和生成日志流程图

不同任务类型，系统调度不同进程。CreateVM PoolTimeJob、CreateVMTimeJob、CreateTemplateTimeJob、CreateMigrationTimeJob 等任务进程，其内部逻辑大致相同。通过实现 Job 接口获得 execute()，实现 LogHandler 记录执行过程中生成的记

录，通过与 RHEVMUtil 的组合向 RHEVM 发送请求。

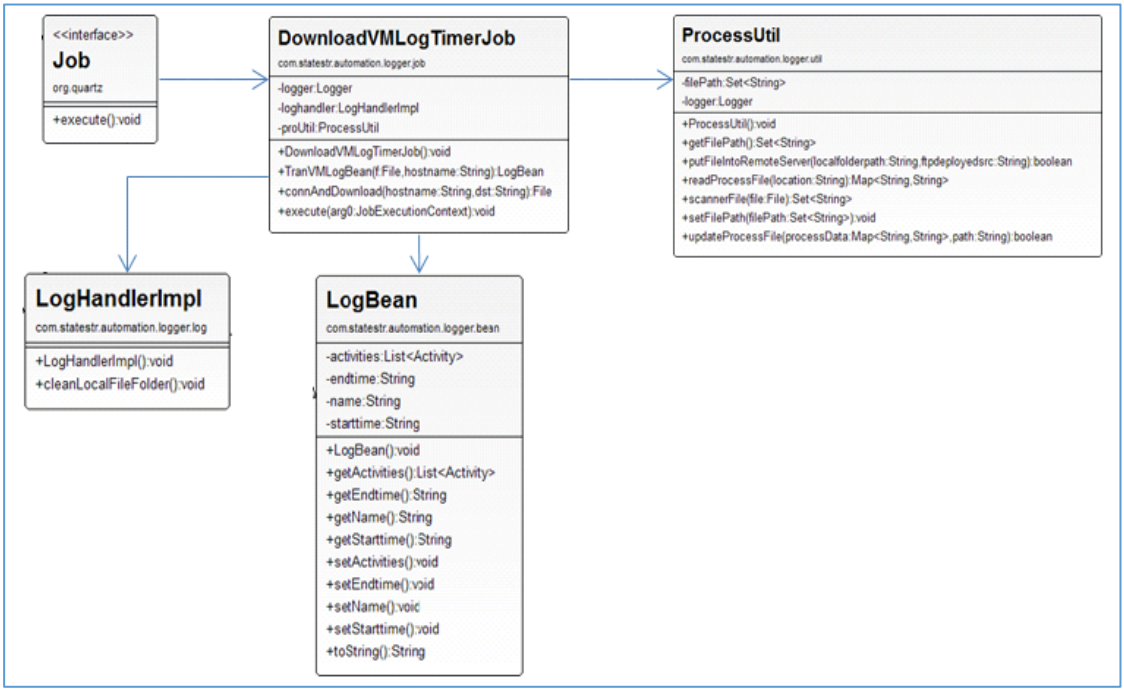

图 4.10 调度进程相关类图

DownlodVMLogTimerJob 依赖于 ProcessUtil、LogBean、LogHandlerImpl 三个类。通过 ProcessUtil 的一个名为 proUtil 的实例得到在指定文件夹下的 VM 的日志文件，合并各个文件形成最终含有日志概要的日志，并更新数据库中。loghandler 是 LogHandlerImpl 的一个实例，负责在任务完成之后将在执行过程中生成临时文件全部清除。LogBean 为 DownlodVMLogTimerJob 提供包装类。DownlodVMLogTimerJob 实现 Job 接口获得 execute()。系统通过继承 DownlodVMLogTimerJob 实现生成 log 对业务逻辑。

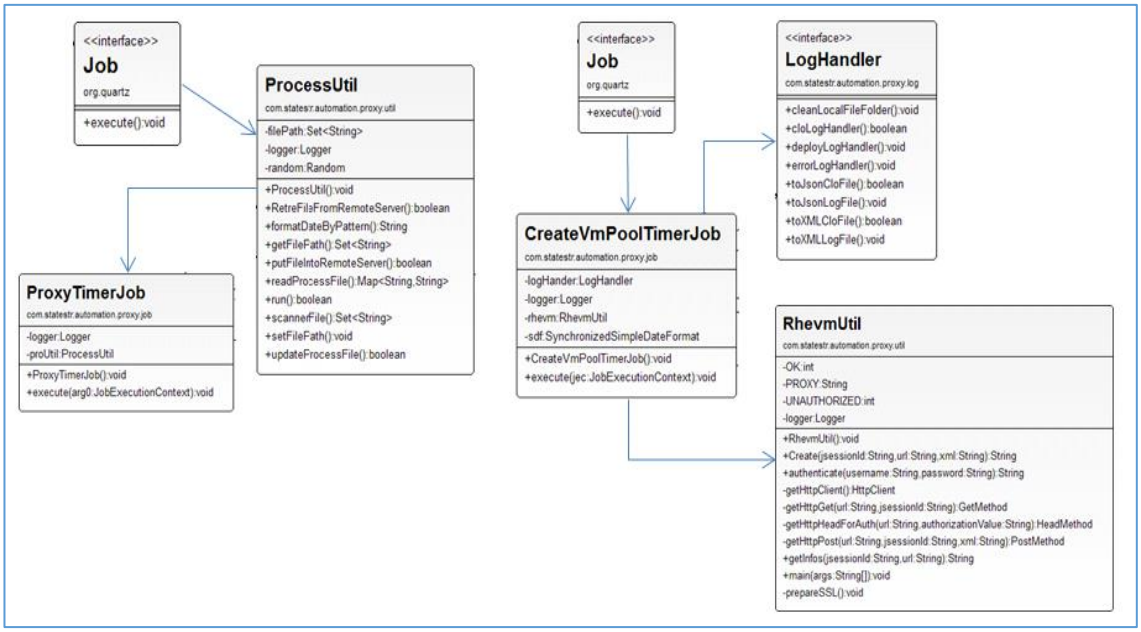

图 4.11 日志生成相关类图

4.3 本章小结

本章主要根据上一章的设计进行实现。首先如何实现自动化系统的整体架构的设计，如何实现数据库的设计，又分别介绍了虚拟机的创建、虚拟机池的创建、模板在不同站点之间的迁移。最后介绍了自动化系统后端是如何实现任务调度和日记生成。

第5章 部署及运行

本章将阐述本自动化管理系统的部署运行所需的软硬件环境，以及在本自动化管理系统部署成功之后的运行结果。

5.1 自动化管理系统的部署

本自动化管理系统正常运行需要一定的软硬件环境等支持。软件环境主要是本自动化管理系统部署运行所依赖的软件服务以及各服务正常运行的依赖软件。本系统前端代码使用 JavaScript 作为开发语言，后端代码以 Java 作文开发语言，所以软件环境方面：RHEV Manager 必须安装在一个基本安装的 Red Hat Enterprise Linux 6.6 或 6.7 系统上。根据 RHEV 安装手册的说明在完成基本环境安装之后，紧接着要安装 RHEVM 环境，不要先安装其它软件，这样是为了保证 RHEVM 不会与其它软件发生冲突。jdk1.6 版本以上，JavaScript 使用的是 ECMASCRIPT 5 版本，浏览器版本要求如下：Chrome 13+、Firefox 4+、Safari 5.1.*、IE 9.*。

表 5.1. RHEV Manager 硬件要求

资源	最小配置	推荐配置
CPU	双核 CPU	4 核 CPU 或多个双核 CPU
内存	4 GB 可用系统内存	16GB 系统内存
硬盘	25GB 本地可写磁盘空间	50GB 本地可写磁盘空间
网络接口	一个带宽最少为 1 Gbps 的网卡（NIC）	带宽最少为 1 Gbps 的网卡（NIC）

虚拟机控制台只能通过 Red Hat Enterprise Linux 系统或 Windows 系统中支持的 Remote Viewer（virt-viewer）客户端进行访问。SPICE 控制台的访问只在其它操作系统上（如 OS X）可用，并只能通过不被支持的 SPICE HTML5 浏览器客户端进行访问。Red Hat Enterprise Linux 和 Windows 7 系统中都包括了支持的 QXL 驱动。

5.1.1 网络浏览器要求

表 5.2 不同支持级别浏览器的版本

支持级别	操作系统	网络浏览器	门户访问
级别 1（Tier 1）	RHEL	Mozilla Firefox Extended Support Release（ESR）版本	管理门户和用户门户

级别 2（Tier 2）	Windows	Internet Explorer 10 或更新版本	管理门户和用户门户
	所有	当前版本的 Google Chrome 和 Mozilla Firefox	管理门户和用户门户
级别 3（Tier 3）	所有	较早版本的 Google Chrome 和 Mozilla Firefox	管理门户和用户门户
	所有	其它浏览器	管理门户和用户门户

用户可以使用以下列出的网络浏览器来访问本自动化系统后端管理系统。在以上的测试组合中只有 Tier3 无法完全支持。所以为部署本自动化管理系统所支持的网络浏览器是版本号满足 Firefox 43.*及以上版本、Chrome 45.*及以上版本、IE 9.* 及以上版本，或其组合。而对于版本过老的浏览器本系统将不做兼容性处理。

5.1.2 操作系统的要求

RHEV Manager 必须安装在一个基本安装的 Red Hat Enterprise Linux 6.6 或 6.7 系统上。所有 CPU 都必须支持 Intel 64 或 AMD64 CPU 扩展，并启用 AMD-V 或 Intel VT 硬件虚拟化扩展。并且需要支持 No eXecute 标识（NX）。所以本系统部署等操作系统选用的较为稳定的 RHEL 7 系统。

5.1.3 内存要求

根据 Guest OS 的要求以及所要安装的应用软件的要求，虚拟机的使用的要求内存的大小的最低限度也无法确定。另外，考虑到 KVM “过度分配”即是分配给虚拟机的内存总量可以大于主机所具有的物理内存总量的情况。KVM 过度分配是基于所有虚拟机不会在同一时间全部使用分配给它们的内存假设条件。KVM 通过只在需要时才为虚拟机分配 RAM 实现这一功能的。

表 5.3 内存要求

最小	最大
2 GB 内存	2 TB 内存

考虑以上所有因素之后，并考虑到未来虚拟机的不断增加等情况，内存设计为 500GB。

5.1.4 存储要求

Hypervisor 主机需要本地的存储设备来保存配置、日志信息、内存 dump 以及交换空间。 RHEV Hypervisor 所需的最小存储配置需求在这里被介绍，而 Red Hat Enterprise Linux 主机所需的存储空间会根据不同情况有所不同，但它们应该会比 RHEV Hypervisor 的存储配置要求更高。

表 5.4 RHEV Hypervisor 版本对存储的配置要求

版本	Root 和 RootBackup 分区	配置分区	日志分区	数据分区	交换分区	最小总计
RHEV Hypervisor 6	512MB	8MB	2048MB	512MB	8MB	3.5 GB
RHEV Hypervisor 7	8600 MB	8MB	2048MB	10240 MB	8MB	20.4 GB

日志（logging）分区不低于 2GB 的存储空间。为保证本自动化管理程序的运行以及考虑到未来系统的扩展，将为其分配 16GB 的存储空间。

对于 Red Hat Enterprise Virtualization Hypervisor 6，数据分区最少需要 512MB 存储空间；对于 Red Hat Enterprise Virtualization Hypervisor 7，数据分区最少需要 1024MB 存储空间。如果还需要安装 RHEVM Virtual Appliance，数据分区最少需要 60GB 存储空间。这个分区的推荐值是内存数量的最少 1.5 倍，再加上额外 512MB。若果数据分区过小将来将无法对 Red Hat Enterprise Virtualization Manager 中主机进行升级，而且 RHEV 的在默认情况是数据分区会占用除去交换空间后的所剩的所有存储空间。在其他分区满足的情况下，分配的数据空间是 2TB 大小。

5.1.5 PCI 设备要求

虚拟主机需要最少一个网卡（最小带宽是 1Gbps）。根据 RHEV 安装手册中的推荐配置是为每台虚拟主机上配置 2 个网卡。其中一个负责虚拟主机的网络连接和通信，另一个专门负责处理需要大量网络数据的操作（如虚拟机的迁移），如果这些操作无法通过网络获得必要的数据，其性能将会受到很大的限制，甚至导致任务失败。

5.2 自动化管理系统的运行

配置好系统环境，就可以运行本自动化管理中心。登录系统要通过公司经理的授权，在账户取得权限之后，通过前端 UI 界面登录系统首页。

图 5.1 自动化管理系统首页

前端页面展示分为图例区和功能区。图例区展示选择站点中的 Host，Virtual Machine， User 的使用情况。功能区分为 View 区和 Create 区。View 区包含 Automation 信息展示，模板信息展示，VM 信息展示，VM Pool 信息展示。Create 区包含 VM Pool 的创建，模板的创建，VM 的创建，迁移的创建。在点击饼图之后会展示对应饼图的详细信息。

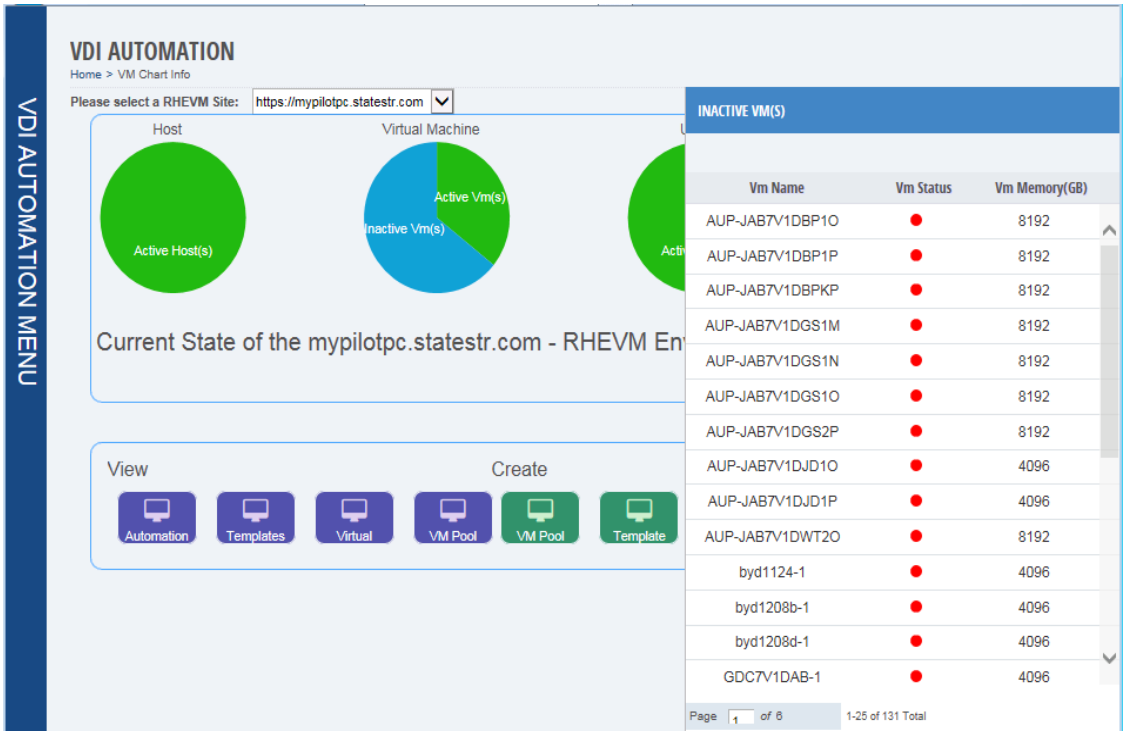

图 5.2 点击查看 VM 详细信息

在 create 功能区点击 VM Pool，用户填写必要的 VM Pool 信息。

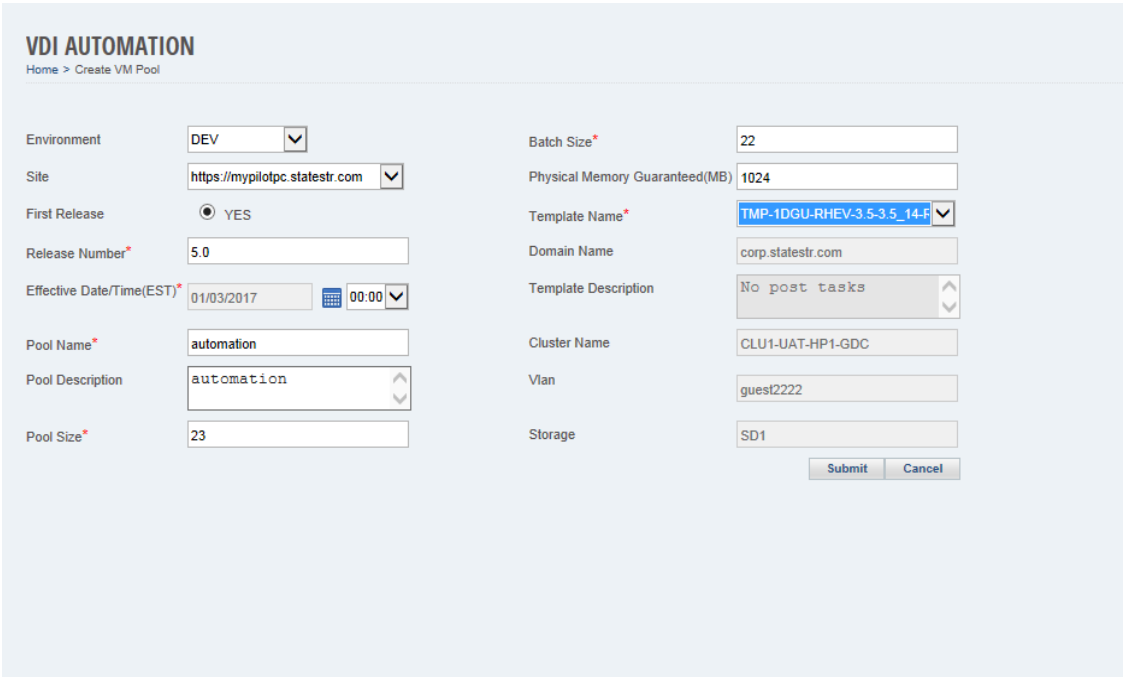

图 5.3 VM Pool 创建页面

在用户填写完整 VM Pool 必要信息之后，点击提交按钮，当页面上出现下面的提示信息时，说明任务已提交完成。

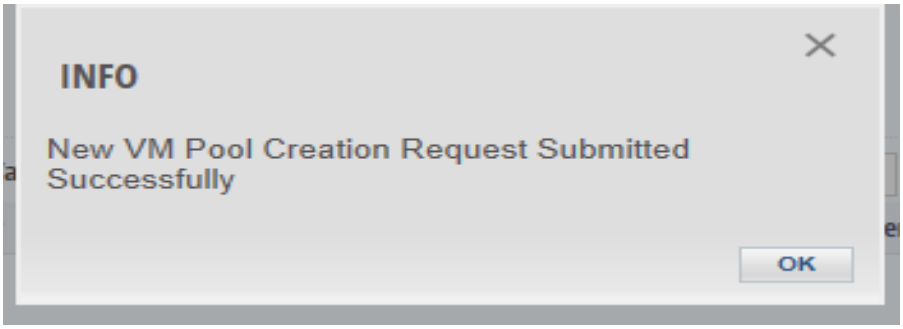

图 5.4 VM Pool 创建请求提交完成提示

在 View 区查看 VM Pool 在会展示出 VM Pool list 页面。在选中一项用户需要编辑的 VM Pool 之后，点击页面中 Edit 功能，系统会弹出 EDIT POOL 功能页面。用户填写完成需要编辑的选项或是需要修改的信息之后点击提交，完成请求。当任务提交完成并成功创建 Edit VM Pool 任务之后，系统会弹出下图的提示信息。

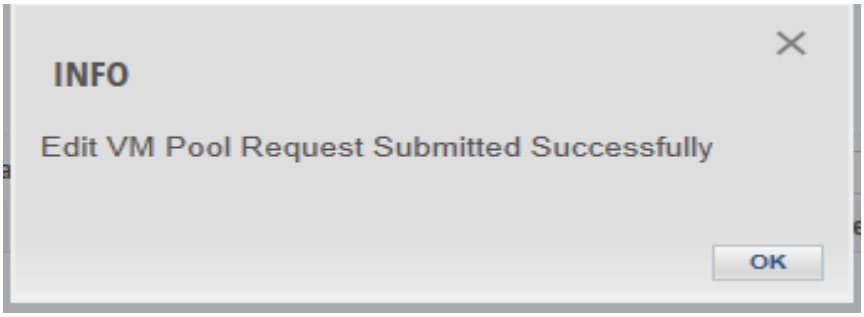

图 5.5 编辑 VM Pool 页面

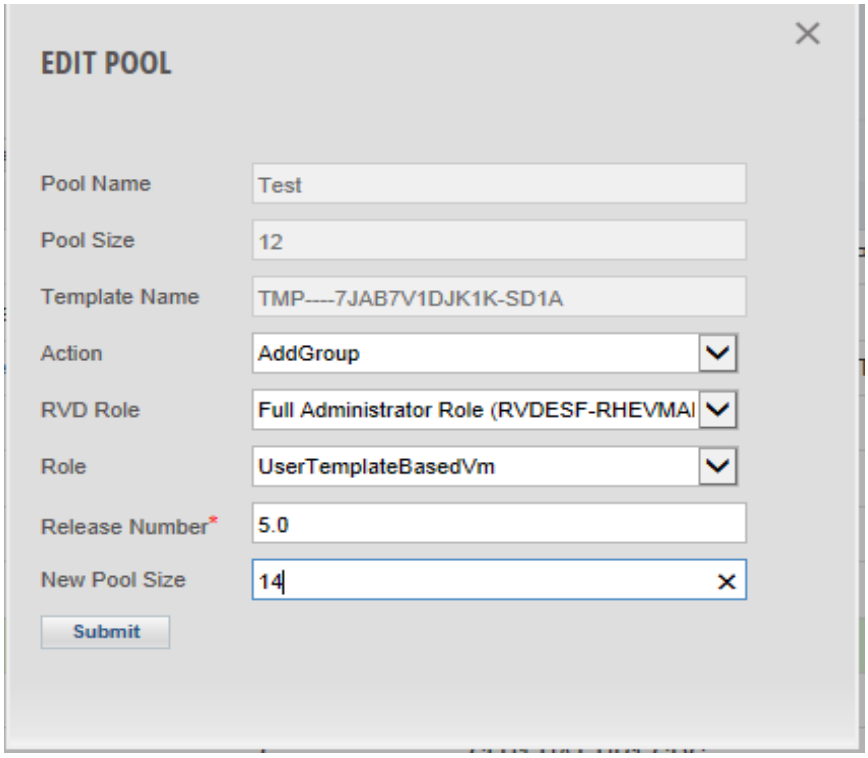

图 5.6 请求编辑 VM Pool 提交成功提示

在 Create 功能区点击模板迁移。界面会跳到如下页面，在此界面选择需要迁移模板，迁移到的站点、存储、集群等信息之后。点击提交。

在任务创建完成之后会出现相应的提示。

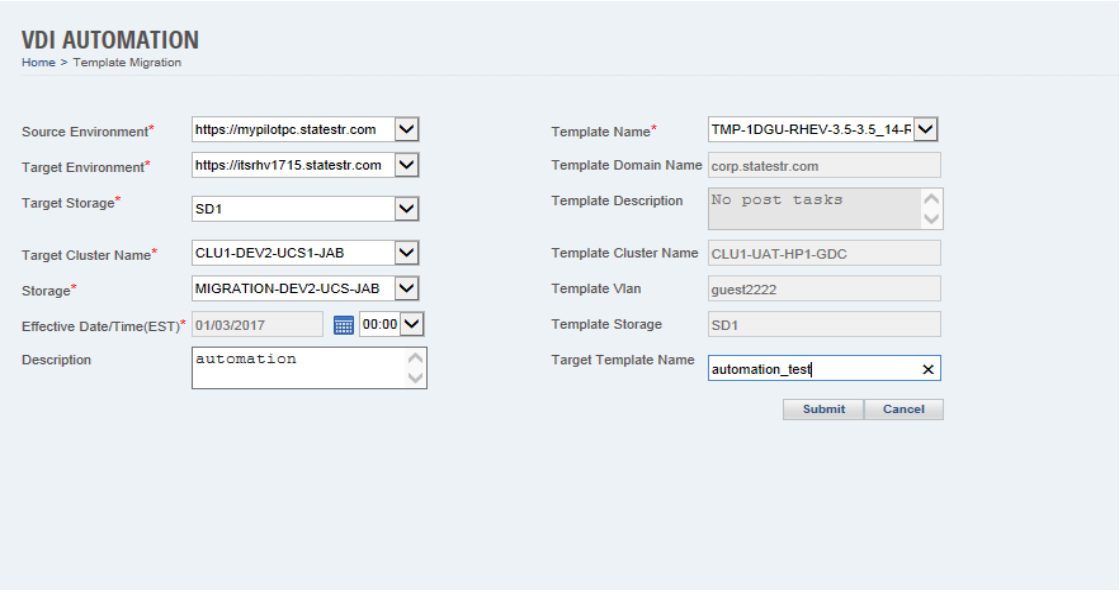

图 5.7 模板迁移页面

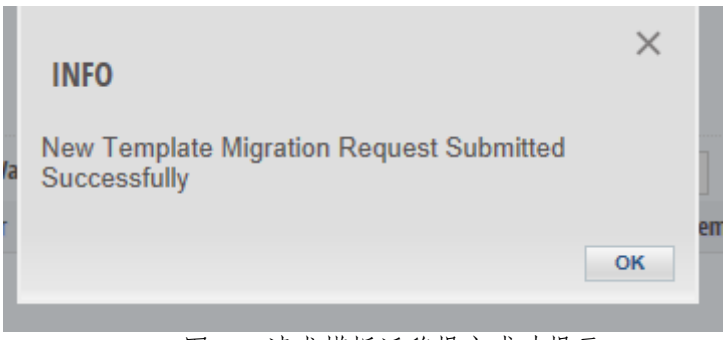

图 5.8 请求模板迁移提交成功提示

在以上任务创建完成之后，在后端和数据库中如果能查到相应的信息就算任务创建完成。在刚刚写入数据的任务是 deploy.log，error.log，clo.log 三个字段值都是默认为空。当数据库中 deploy.log，error.log，clo.log 的值发生改变并有相应的信息时说明任务已经被执行，可以在 RHEVM 中查到相应的信息。

TID	EXT_APP_CODE	VERSION	CREATED_BY	CREATED_TMS	OPERATION_TYPE	DEPLOY_STATUS	ERROR_LOG
TID_VAFWEB83285_1.22_VAF000352	VAFWEB83285	1.11	A560000	01-03-2017 00:38	EditVMPool	2	{ "starttime": "01/03/2017 00:40:59", "endtime": "01/03/20
TID_VAFWEB83285_1.22_VAF000354	VAFWEB83285	2.11	A560000	01-03-2017 01:00	EditVMPool	2	{ "starttime": "01/03/2017 01:02:59", "endtime": "01/03/20
TID_VAFWEB83285_1.22_VAF000353	VAFWEB83285	2.33	A560000	01-03-2017 00:50	EditVMPool	2	{ "starttime": "01/03/2017 00:52:59", "endtime": "01/03/20
TID_VAFWEB83285_1.22.21_VAF000356	VAFWEB83285	1.22.21	A560000	01-03-2017 01:10	VMPool	2	{ "starttime": "01/03/2017 01:12:59", "endtime": "01/03/20
TID_VAFWEB83285_0_VAF000359	VAFWEB83285	(null)	A560000	01-03-2017 22:35	Template Migration	1	(null)
TID_VAFWEB83285_5.0_VAF000358	VAFWEB83285	5.0	A560000	01-03-2017 22:33	VMPool	1	(null)
TID_VAFWEB83285_1.22_VAF000355	VAFWEB83285	1.23.3	A560000	01-03-2017 01:05	EditVMPool	2	{ "starttime": "01/03/2017 01:07:59", "endtime": "01/03/20
TID_VAFWEB83285_1.33.312_VAF000357	VAFWEB83285	1.33.312	A560000	01-03-2017 01:21	VMPool	2	{ "starttime": "01/03/2017 01:23:59", "endtime": "01/03/20

DEPLOY_LOG	CLO_LOG	PROCESS_CONTROLLER
{ "starttime": "01/03/2017 00:40:59", "endtime": "01/03/2017 00:41:09",...	"logs" : [{ "starttime": "01/03/2017 00:40:59", "endtime": "01/03/2017 ...	ENVIRONMENT=UAT|RHEVM=https://r
{ "starttime": "01/03/2017 01:02:59", "endtime": "01/03/2017 01:03:00",...	"logs" : [{ "starttime": "01/03/2017 01:02:59", "endtime": "01/03/2017 ...	ENVIRONMENT=UAT|RHEVM=https://r
{ "starttime": "01/03/2017 00:52:59", "endtime": "01/03/2017 00:53:00",...	"logs" : [{ "starttime": "01/03/2017 00:52:59", "endtime": "01/03/2017 ...	ENVIRONMENT=UAT|RHEVM=https://r
{ "starttime": "01/03/2017 01:12:59", "endtime": "01/03/2017 02:06:08",...	"logs" : [{ "starttime": "01/03/2017 01:12:59", "endtime": "01/03/2017 ...	ENVIRONMENT=UAT|RHEVM=https://r
(null)	(null)	SOURCESITEURL=https://mypilotp
(null)	(null)	ENVIRONMENT=DEV|RHEVM=https://r

图 5.9 任务为执行之前的数据库中信息

TID	EXT_APP_CODE	VERSION	CREATED_BY	CREATED_TMS	OPERATION_TYPE	DEPLOY_STATUS	ERROR_LOG	DEPLOY_LOG
TID_VAFWEB83285_0_VAF000359	VAFWEB83285	(null)	A560000	01-03-2017 22:35	Template Migration	3	{ "starttime": "01/03/2017 22:37:59", "endtime": "01/03/2017 22:38:25",...	{ "starttime": "01/03/2017 22:3
TID_VAFWEB83285_5.0_VAF000358	VAFWEB83285	5.0	A560000	01-03-2017 22:33	VMPool	1	{ "starttime": "01/03/2017 22:35:59", "name": "error.log", "activities...	{ "starttime": "01/03/2017 22:3
TID_VAFWEB83285_4.0_VAF000360	VAFWEB83285	5.0	A560000	01-03-2017 22:39	EditVMPool	1	{ "starttime": "01/03/2017 22:41:59", "name": "error.log", "activities...	{ "starttime": "01/03/2017 22:4

DEPLOY_LOG	CLO_LOG	PROCESS_CONTROLLER
{ "starttime": "01/03/2017 22:37:59", "endtime": "01/03/2017 22:38:25",...	"logs" : [{ "starttime": "01/03/2017 22:37:59", "endtime": "01/03/2017 ...	SOURCESITEURL=https://mypilotpc.statestx.com|DESTINATIONSITEURL=
{ "starttime": "01/03/2017 22:35:59", "name": "deploy.log", "activitie...	"logs" : [{ "starttime": "01/03/2017 22:35:59", "name": "deploy.log", ...	ENVIRONMENT=DEV|RHEVM=https://mypilotpc.statestx.com|FIRSTPELEAS
{ "starttime": "01/03/2017 22:41:59", "name": "deploy.log", "activitie...	"logs" : [{ "starttime": "01/03/2017 22:41:59", "name": "deploy.log", ...	ENVIRONMENT=DEV|RHEVM=https://mypilotpc.statestx.com|FIRSTPELEAS

图 5.10 任务被调度执行之后数据库的信息

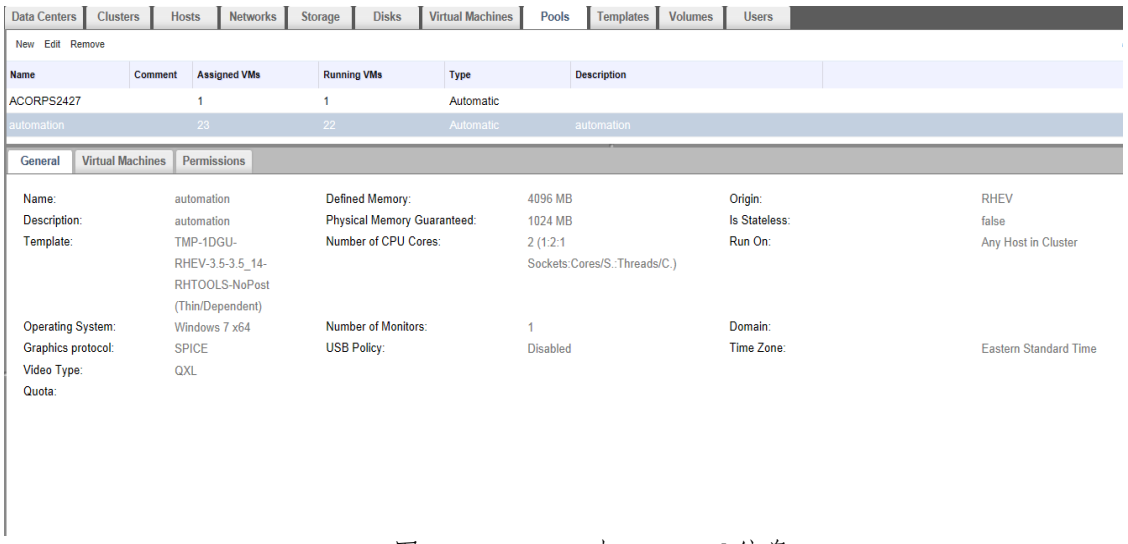

图 5.11 RHEVM 中 VM Pool 信息

5.3 本章小结

本章首先对自动化管理系统部署运行所需要的软硬件环境进行基本的介绍，本系统运行需要的基本环境要求，包括网络环境要求、操作系统环境要求、内存的要求、存储容量的要求、PCI 设备的要求等。其次，对本自动化管理系统进行了测试运行。主要测试了查看 VM 的详细信息和 VM Pool 的创建，模板的迁移等功能并保存展示来运行测试的结果截图。截图证明了本自动化管理系统部署运行成功，基本实现了在本文设计的功能点，满足来用户的基本需求。

第6章 总结与展望

本章主要对本文设计的自动话管理系统的工作进行总结并提出一些改进建议，之后对虚拟化技术未来的发展方向进行一定的展望。

6.1 本文总结

虚拟化技术为企业提高硬件利用率，降低耗损，节约成本以及降低了业务运营的难度等优势，在行业中取得普遍的认可和广泛的应用。尽管像 RHEV 这样的虚拟化平台稳定、高效、可扩展，但对于使用者来说其背后的业务逻辑相对繁琐。本文针对这点主要做了以下工作:

1、自动化创建虚拟机，用户只需要输入虚拟机名称，选择要使用模板，Cluster 等必要信息系统会自动创建虚拟机，并执行 PowerShell Script 为安装 OS 作好准备。

2、自动化创建虚拟机池，用户输入虚拟机池的名称，选用使用的模板，指定池的大小等，系统自动创建虚拟机池，并按照用户选用的模板中的配置信息和指定虚拟机池大小来创建虚拟机，在创建虚拟机完成之后，由后端任务执行模块请求为每个虚拟机分配 IP 地址。

3、自动化创建模板，用户选择源虚拟机、域、集群、角色和权限等信息，系统自动创建模板。

4、自动化迁移模板，用户选择源站点，源站点中可用的模板并选择模板迁移的目标站点，设定模板在目标站点使用的 Cluster，Storage 等信息，系统自动迁移模板。

5、查看 VM、VM Pool、模板、模板迁移的基本信息和日志。

本系统经过测试运行基本达到用户的需求。由于整个项目功能繁多、业务逻辑极其复杂、涉及的领域较多，在设计和实现中难免有所疏忽和遗漏。可以本文设计实现的自动化管理系统作如下的改进: 1、在业务处理过程中，业务逻辑单元需要发送多次 RPC 请求获得必要数据。数据在业务完成之后被释放，下此再处理其他业务时会再次请求此数据。在系统里加入缓存机制，可有效节约时间和缓解网络压力。2、本系统的用户包含分别在不同国家，可以将系统字符统一在一起形成字符集，针对不同的国家设计不同的字符集。各个国家的用户根据自己的习惯使用不同的字符集。3、本系统并发程度还不高，将会在未来的使用中受到限制。可以使用高并发、多线程技术进行重写已满足未来更多用户的使用。4、本系统中支持定时任务，在用户在制定未来某个时间点执行任务之后，不能实时查询任务是否得到执行。可以在用户的定时任务得到执行指挥，给予邮件提醒或事短信提醒用户。

6.2 展望

虚拟化技术发展朝着两个完全不用的方向[48]: 一个是朝着纯软件虚拟化发展防线; 一个朝着硬件虚拟化发展方向。本文设计的自动化管理系统是以纯软件方

式来实现的，本系统也需要硬件虚拟化的支持才能完成部署运行。随着硬件辅助虚拟化的不断发展，相信虚拟化硬件将会突破当前硬件的瓶颈，以全新架构方式来支持虚拟化指令，虚拟机管理器可以直接运行在硬件设备上，同时 VMM 架构将会被重新设计[49]，从而极大简化 VMM，执行效率极大提高，VMM 也将更加高效，更加标准化。可以想象，GPU 虚拟化的发展将会直接威胁到个人计算机的使用，能使虚拟化得到更加全面的普及。软件虚拟化的发展将与传统的软件相互结合而发展形成不同的虚拟化分支，各个方向的虚拟化产品命名也会不一样，如虚拟化与云结合发展为桌面虚拟化 、与系统架构平台服务结合发展为平台虚拟化、与数据库服务结合发展为数据库虚拟化等等，类似还有应用程序虚拟化、网络虚拟化等虚拟化发展方向[50]。随着虚拟化技术发展，两个不同的虚拟化发展最终会结合在一起形成系统化、结构化、组织化的虚拟化技术，这种技术或许会成未来计算机底层最基本的技术之一。 虚拟化技术与云计算的结合，将会最大限度的充分利用现有的硬件资源，最大限度提升效率，并节约成本。

参考文献

[1]华为科技有限公.FusionSphere 5.0 虚拟化技术白皮书[R], 2014
[2]马璟.基于 VMware 技术的服务器虚拟化架构的研究与应用[D].硕士研究生,厦门大学 2011
[3]Rangan K, Cooke A, Post J, Schindler N.The Cloud Wars: 100+billion at stake.Analyst The，[J].No.May, 2008:1－90
[4]Siegele L.Let It Rise:A Special Report on Corporate IT[J].The Economist, No.October, 2008:1-14
[5]R.P.Goldberg. Survey of Virtual Machine Research[J].IEEE Computer Magazine, 1974:34-45
[6]Nanda, S., Chiueh.T.A survey on virtualization technologies[EB/OL]. 2005, http://www.ecsl.cs.sunysb.edu/tr/TR179.pdf
[7]李亚琼,宋莹,黄永兵.一种面向虚拟化云计算平台的内存优化技术[J] 中国科学院 100190.3724/SP.J.1016.2011.00684
[8]Gerald J.Popek, Robert P.Goldberg, Formal requirements for virtualizable third generation architectures[J], Communications of the ACM, v.17 n.7, p.412-421, July 1974
[9]Sean Capbell, Michael Jeronimo An Introduction to Virtualization[R] Intel Corporation 2006
[10]vmware white paper, virtualiztion Overview[R].vmware inc.2006
[11]Morty Eisen.Introduction to Virtualization The Long Island Chapter of the IEEE Circuits and Systems (CAS) Society [R],Apri l28th, 2011
[12]陈昌峰.虚拟机服务器技术在网站建设中的应用[D].硕士研究生,复旦大学 2012 TP393.092
[13]黄峰.分布式虚拟运行环境的研究与实现[D].硕士研究生,国防科学技术大学.2007
[14]李永.基于虚拟机动态迁移技术的分析和研究[D].硕士研究生,国防科学技术大学 2007
[15]高小明.基于 Intel VT 硬件虚拟机内核研究与实现.电子科技大学 2010
[16]杜旭生.目前服务器虚拟化技术应用的现状[EB/OL].2010-04-08 http://www.aixchina.net/Article/20265
[17]EricFeagler.微软虚拟化技术的独到之处[R].软大中华区服务器产品业务群 2009 年
[18]王晓静.I/O 虚拟化的性能隔离和优化[D].博士研究生,华中科技大学.2012
[19]胡冷非.虚拟机 Xen 网络带宽分配的研究和改进[D].硕士研究生,上海交通大学 2009
[20]邵文清.基于 Xen 的云管理平台下资源调度策略的研究与实现[D].硕士研究生,西安电子科技大学 2012
[21]高清华.基于 Intel VT 技术的虚拟化性能研究[D].硕士研究生,浙江大学 2008 年 5 月
[22]虚拟化技术[EB/OL].[2012-11-23]https://zh.wikipedia.org/zh-cn/虚拟化
[23]李蓯蒲.基于 Intel VT 技术的 PC 虚拟化平台研究与测试[D].硕士研究生,北京邮电大学 2008
[24]林昆.基于 Intel VT-d 技术的虚拟机安全隔离研究[D].硕士研究生,上海交通大学 2011
[25]杨柳青.硬件虚拟机 Xen 的研究和性能优化[D].硕士研究生,浙江大学 2008
[26]刘可超.基于 Xen 的虚拟存储系统的研究和改进[D].硕士研究生,上海交通大学 2010
[27]朱鸿伟.虚拟化安全关键技术研究[D].硕士研究生,浙江大学 2008
[28]曹欣.半虚拟化技术分析与研究[D].硕士研究生,浙江大学 2008
[29]马喆,禹熹,袁傲等 Xen 安全机制探析[J] 信息网络安全 2011(11):31-35
[30]LeVasseur Joshua.Uhlig Volkmar.Chapman Matthewet al.Pre-virtualization:Slashing the cost of virtualization NICTA: Technical Report PA00520.2005
[31]温研.隔离运行环境关键技术研究[D].博士研究生,国防科学技术大学 2008
[32]程霖.KVM 虚拟机在线迁移性能优化的研究与实现[D].硕士研究生,华中科技大学 2014
[33]叶可江.虚拟计算系统的性能和能耗管理方法研究[D].博士研究生,浙江大学 2013
[34]李传云.KVM 虚拟机热迁移算法分析及优化[D]. 硕士研究生,浙江大学，2016.01.09
[35]郭晋兵.吴超凤.虚拟机迁移技术漫谈[EB/OL] .[2010-09-09].http://www.ibm.com/developerworks/cn/linux/l-cn-mgrtVM1/

[36]施杨斌.云计算环境下一种基于虚拟机动态迁移的负载均衡算法[D].硕士研究生,复旦大学 2011
[37]杨寒冰.虚拟机动态迁移技术的研究[D].硕士研究生,南京邮电大学 2013
[38]王仕象.关于虚拟机迁移技术的应用研究[J].科技传播 2012(20):1
[39]杨寒冰.虚拟机动态迁移技术的研究[D].硕士研究生,南京邮电大学 2013
[40]邹潇.基于脏页率预测的虚拟机动态迁移研究[D].硕士研究生,上海交通大学 2012
[41]吴官林.高可用性虚拟化管理中心的设计与实现[D].硕士研究生,西安电子科技大学 2011
[42]李雨桐.高可用性虚拟化管理框架的研究与实现[D].硕士研究生,东北大学 2013
[43]冉彬作.轻量级目录访问协议（LDAP）在大学资源计划（URP）中的应用[D].硕士研究生,重庆大学 2005
[44]陈洪涛,齐鸣,唐屹,丁建立.基于 LDAP 的空管用户目录服务系统构建[J] 计算机工程与设计 2010(19):4205—4208
[45]陈跃武.万晓冬.LDAP 在仿真资源数据网格中的应用研究[J] 计算机仿真 2007(10):119—122
[46]朱建,周爱霞,卢晨晨,罗炜,沈晔.基于目录服务的语义 Web Service 共享技术[J] 江南大学学报（自然科学版）2013(1):50—53
[47]孙骏.面向登录管理的企业安全认证集成[D].硕士研究生,复旦大学 2006
[48]Yoshihiko Oguchi.Tetsu Yamamotoserver.Server Virtualization Technology and Its Latest Trends[J].FUJITSU Sci.Tech.2008-1(1):46-52
[49]陈文智,姚远,杨建华,何钦铭.Pcanel/V2--基于 Intel VT-x 的 VMM 架构[J].计算机学报.2009-7(7):1311-1319
[50]胡晓荷.虚拟化时代即将来临[J] 信息安全与通信保密 2009(3):14—17

作者简历

1990 年 10 月出生于河南省郸城县，先后在村中的小学和镇上的初中学习，随后选入郸城县第一高级中学完成高中教育。2009 年 9 进入鞍山师范学院数学系（后改为数学与信息科学学院）学习计算机科学与技术专业，于 2013 年完成本科教育，并获得工学学士学位。2013 年第一次考研，报考浙江大学计算机学院，因复试未通过心有不甘，2014 年再度报考浙江大学，4 月被浙江大学软件学院录取，7 月份进入浙江大学软件学院宁波校区学习，并加入超大规模信息系统研究中心，于 9 月正式开始攻读研究生课程。2015 年 5 月到浙大网新恒天科技有限公司进行实习，同月外派到美国道富科技（杭州）公司。在此期间申请加入了浙江大学创新与创业管理强化班（ITP），因此有机会于 2016 年暑假期间到美国硅谷 mProbe 大数据健康公司参加了实习。

致谢

研究生阶段的学习就要结束，在这两年半里，我不仅仅学到了更加专业的知识，接触到了专业最前沿的科技技术，也从老师和同学们身上感受到了浙大人务实求是、竭力创新的精神品格。

首先感谢我的导师周波教授和合作导师尹可挺博士。周波教授在我实习期间给予我最大的支持和关照，使我能顺利的完成实习期间各项任务，充分锻炼了我各方面的能力。我十分敬佩周波教授。周波教授以其丰富创业经历和企业管理经验，全面的诠释了浙大人求是创新精神。是我学习的标杆和榜样。

尹可挺博士我的整个研究生期间在学习上给予了最大的指导和帮助。他严谨认真学术态度是深刻影响了我对研究生生活的认知。尹可挺在高科技技术的不断探索、并创建了新科技技术公司，更让我对他更加的佩服。他的人生经历将激励我对未来生活的不断的探索。

其次，我要感谢学校和同学们。感谢同学提供的教学平台和教学资源，让我顺利的完成本论文。感谢同学们一起相互支持、相互帮助、相互陪伴一起完成本次毕业论文。同时，感谢论文评审的各位老师们，感谢你们百忙之中认真指正我论文中的不足。

再次感谢家人和亲人们。感谢他们给予精神上的支持和陪伴。因为有他们的支持，我能一往无前追逐我想要的人生。因为有他们的支持，我感到在前行的道路不是孤独的。因为有他们的支持，我心中充满牵挂和恋念。家人中特别要感谢的是生活在那边的她，希望没有辜负你的期望，没有让你失望—这一直是我最怕的事情。

最后要感谢的是自己。感谢自己没有放弃梦想，感谢自己对生活充满憧憬和渴望。

陈家星

于浙江大学软件学院

2017 年 1 月 10 号