清华大学

综合论文训练

题目:基于虚拟机的LIGO软件定制服务

系别: 自动化系

专业: 自动化

姓名: 李紫阳

指导教师: 曹军威研究员

2010 年 6 月 20 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：李紫阳 导师签名：[illegible] 日期：2010.7.1

中文摘要

为了探测引力波，美国建立了 LIGO (Laser Interferometer Gravitational Wave) 天文台和相应的科学协作组织 LSC (LIGO Scientific Collaboration)。LIGO 设备运行产生大量的数据，需要大量定制化的科学应用软件来进行分析。本文的目的是为 LSC 的科学家们提供一种简易的软件定制解决方案，设计并实现了一个可以前端网页得到用户定制的软件信息，后台可自动根据用户需求利用虚拟机进行软件安装的系统。论文首先介绍了 LIGO 项目背景和工作原理，并介绍了系统用到的几款 LIGO 专业软件的基本情况和安装方法。然后，本文对 Linux 软件管理方案进行了分析，根据项目的特点选择虚拟机作为软件定制的载体，给出较为详细的系统需求分析，并介绍了设计实现本系统所需要用到的几种工具。最后，本文给出了 LIGO 软件虚拟机定制系统的具体实现和各脚本流程。

关键词：LIGO；软件定制；自动安装；虚拟机；动态扩展

ABSTRACT

For the detection of gravitational-wave, USA founds the LIGO(Laser Interferometer Gravitational-Wave Observatory)and the LIGO Scientific Collaboration (LSC) at the same time. There is a mass of data produced which need some technical software tools to analyze while LIGO devices are running. The purpose of this graduation design is to design and realize a system that includes a Web server collects the software customize information from user in front, install software automatically take advantage of virtual machine base on the requirement in background. Firstly, this thesis introduces the background of LIGO project, the working principle of LIGO device, introduces some LIGO software which used in the system and presents the installation procedures of the LIGO software. Then, the thesis analyzes the methods exist for the Linux software management, chooses to use the virtual machine as the supporter of software management, presents the details of system's requirement analysis, and introduces a few kinds of tools which used in the system. At last, this thesis gives the realization of LIGO software virtual machine customize system which is build up in this graduation design.

Keywords： LIGO; software customizing; automatically installation; virtual machine; dynamically extend

目录

第1章 引言

1.1 LIGO 简介

1.1.1 LIGO 产生的背景[1]

爱因斯坦在 1916 年发表的广义相对论中预言了引力波的存在。他将空间和时间描述为现实的不同相位，它们在物质和能量本质上是一样的。时空可以被想象为可以用尺子测量距离和用钟表测量时间的一种“布料”。巨大的质量或能量会使时空发生扭曲——就像布料发生变形一样——我们以重力的形式观测到这种扭曲。自由下落的物体，不论是足球，卫星或者是星光，都是简单地沿着这个扭曲时空中最短的路径前进。

当突然发生剧烈运动时，这个时空的曲率以波动形式向外变化，类似于不平静的池塘表面上的涟漪。想象一下两个中子星沿着互相围绕着的轨道转动。中子星是恒星爆炸后通常会遗留下来的产物，是一种具有令人难以置信的密度的天体。它通常在几千米的范围内就包含有可相比于恒星的质量。当两个如此高密度的天体互相围绕运行时，时空被它们的运动搅动起来，引力能量呈波状在宇宙中传播。

1974 年， Joseph Taylor 和 Russell Hulse 在银河系找到了这样的一对中子星。其中一个天体为脉冲星，意味着它向地球辐射有规律的电磁脉冲。Taylor 和他的合作者们像使用一个非常精确的时钟一样使用这些电磁脉冲，以了解中子星的轨道运动。20 多年后，这些科学

图1.1 引力波形成示意图

家发现这些脉冲的有非常奇妙的偏移，这预示着沿轨道运行的天体有着某种能量损失——就是被引力波带走的能量。这个结果恰恰是爱因斯坦的理论所预言的。

自古以来，人类主要依赖探测不同形式的光来观察宇宙。现在，我们处于天文学的前沿——引力波天文学。引力波包含宇宙中天体的运动信息。由于在大爆炸的瞬间和光出现的很久之前，宇宙对引力波是透明的，引力波允许我们比以往更远地向前探测宇宙的历史。并且引力波不会被宇宙中的其他物质吸收或者反射，可以观察到他们被创建时的形式。最重要的是，引力波有可能包含有未知的信息。每当人类向宇宙睁开“眼睛”，总是会发现一些超乎想象的某些事物彻底改变我们看待宇宙方式。现在通过美国的引力波探测器（LIGO）和它的国际合作者，人类准备通过一系列新型的不依赖于光的“眼睛”来看这个宇宙。

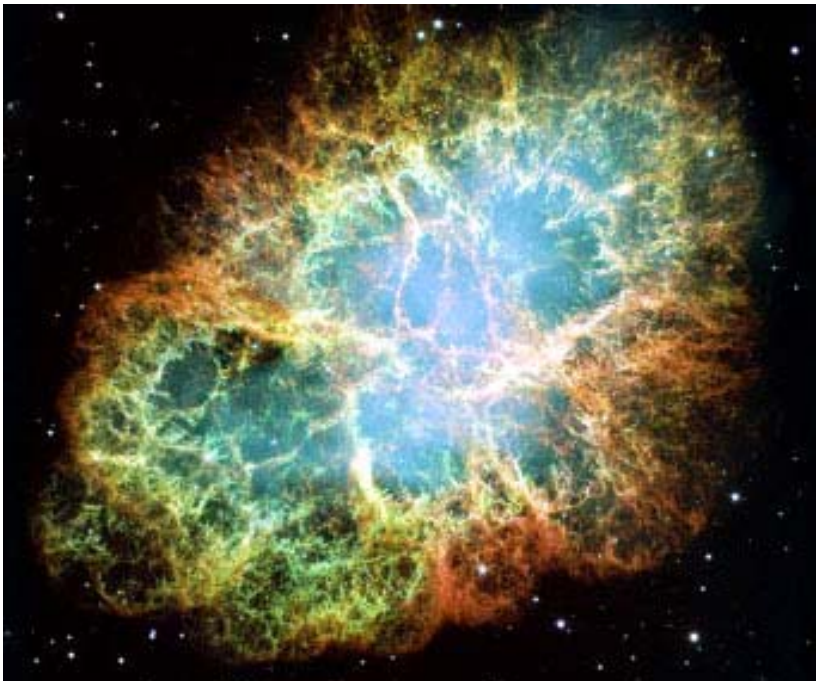

图1.2　　蟹状星云

蟹状星云是 1054 年被人类发现的由一颗超新星的残骸形成的星云。在这个星云的中心是一个脉冲星（一种中子星）为恒星爆炸残留物。超新星爆炸和脉冲星均为引力波的潜在来源。

1.1.2 LIGO 的工作原理

LIGO 使用一种称为激光干涉仪的设备来探测时空中的波动，这种仪器通过使用可控的激光来高精度地测量光在两个悬浮的镜子之间传播的时间。两面镜子悬挂的距离很远，形成干涉仪的一个“臂”，另外有两面镜子成为与第一个臂成直角的第二个臂。由此可见，两个臂构成一个“L”形，如**错误！未找到引用源。**所示。激光通过在 L 的角处的分光镜分别进入两个臂。这束光在两面镜子之间反复反射直到返回分光镜。如果两个臂长度一样，那么返回分光镜的光束发生的干涉会显示出所有的光线返回了激光器。但是如果两个臂的长度有一点不同的话，那么一些光线会传播到别处，而被光电探测器探测到。

当引力波通过的时候，时空波动会引起光束测量的距离发生变化，进而引发落到光电探测器上的光不同。光电探测器将生成一个定义为落到它上面的光随时间变化的信号。LIGO 建造了三个这样的干涉仪，两个位于华盛顿州里奇兰附近，另一个位于路易斯安那州巴吞鲁日附近。LIGO 需要至少两个同步操作、分隔很远的探测器，以排除错误的信号和确定引力波是否通过地球。

图1.3 华盛顿州汉福的 LIGO

图1.4 路易斯安那州利文斯顿的 LIGO

LIGO 激光干涉引力波天文台于 1999 年 11 月建成，它由国家科学基金（NSF）提供资金，耗资 3.65 亿美元。LIGO 是国家科学基金所支持的最大也是最野心勃勃的项目，将同时作为一种国家资源同时为物理学和天文学的科学家们服务。LIGO 科学协作组织（LSC）[3]是由一组科学家组成的，他们以直接对引力波的直接观测作为第一手资料，使用它们探索引力的物理基础，并且将引力波科学发展为天文学探索的新兴领域。LSC 致力于在引力波探测技术的基础上进行研究和发

展，开发并运行引力波探测器。LIGO 是目前全世界最大的、灵敏度最高的引力波探测所，LSC 的一系列的升级计划将更进一步提高其灵敏度。2005 年，激光干涉引力波天文台开始进行了包括采用更高功率的激光器、进一步减少振动等改造。改造之后的探测器灵敏度将提高 1 个数量级，称为先进激光干涉引力波天文台（Advanced LIGO）。

1.1.3 LIGO 软件

目前 LIGO 引力波脉冲数据分析的挑战之一来源于对在线实时探测的需求，如能在短时间内完成海量数据的采集、整理、处理与分析，将为进一步的传统天文观测提供宝贵的准备时间，进而形成全新的天文观测和天体物理研究方法。LIGO 设备运行产生大量数据，有一系列专门的数据分析和处理软件。

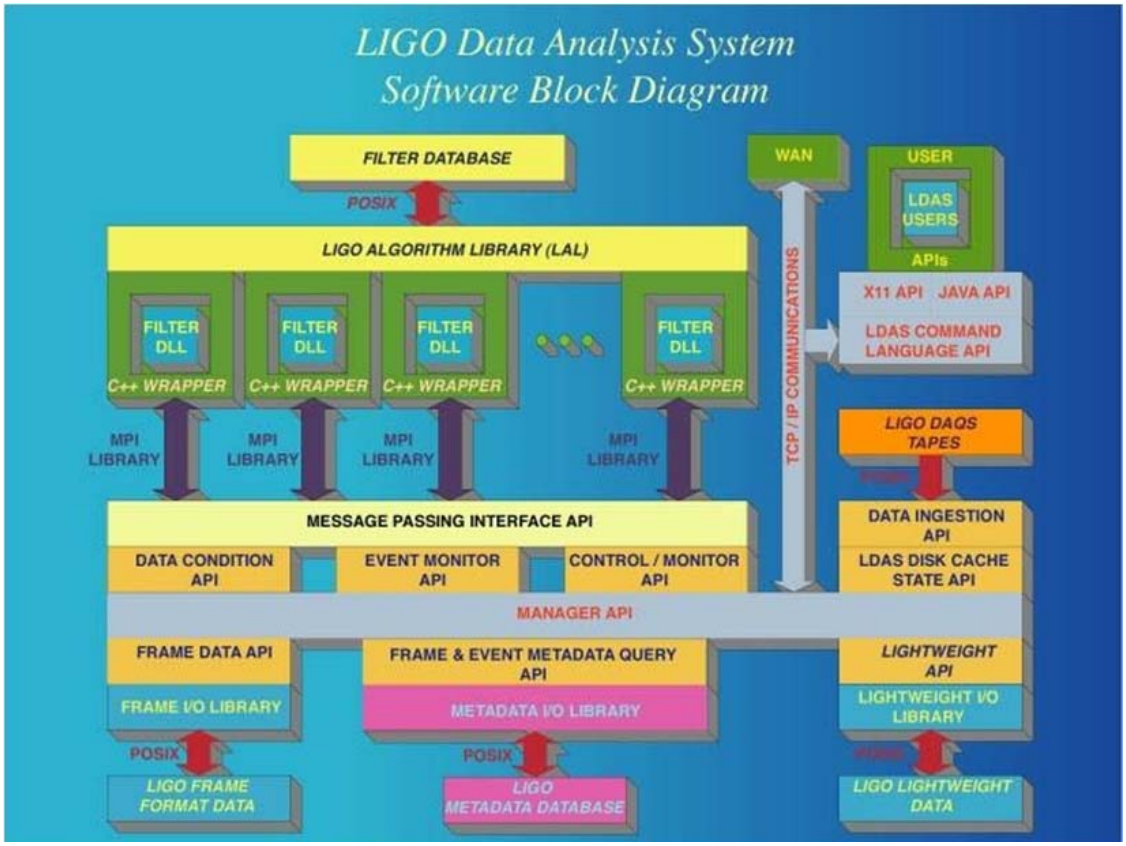

图1.5　　LIGO 数据分析系统

LIGOtools[5]

LIGOtools 是 LIGO/LSC 中专门负责数据文件处理的软件工具包，LIGOtools 的一个基本特点是把所有的可执行程序放在一个单独的 bin 文件夹中，所有的用户链接库放在一个单独的包含所有的包含文件的 lib 文件夹中，所有的 Matlab 脚本放在单独的 matlab 文件夹中等等（实际上是真实的文件由各个包所保有，用符号链接来使所有东西出现在一个单独的文件夹中）。这样，只增加一个 LIGOtoolsbin 文件夹到 PATH 环境变量中可以访问所有现有的和未来的 LIGOtools 程序。LIGOtools 提供一个叫做 ligotools_update 的脚本来检查所有的新包或现有包的最新版本，并且自动下载和安装它们，所有日常管理这些软件是很简单的。

LIGOtools 的安装过程为：

- 下载 *ligotools_init_2.4.tar*
- 解压缩 *tar xmf ligotools_init_2.4.tar*
- 执行*./ligotools_init*
- 设置环境变量 *eval `<dir>/bin/use_ligotools`*
- 下载 *tclexe_8.4.7_LINUX.tar.gz*
- 执行 *ligotools_install tclexe_8.47_LINUX.tar.gz*
- 执行 *ligotools_update*

表 1.1 为执行 ligotools_update 时所安装的软件包。

表1.1　　ligotools_update 命令中安装的包

Package	Description
Fr	C library and utilities in core Virgo distribution to read/write data in frame format
FrContrib	Additional utilities for working with frame files, exclusive of core Virgo distribution
dataflow	Raw data and metadata access utilities
detgeom	Matlab routines to define and manipulate detector geometry
guild	Graphical User Interface to LIGO Databases
httptools	Simple utilities to retrieve files via http
ilwdread	Matlab script to read an ilwd file
ldasjob	High-level interface for running LDAS jobs from Tcl scripts

medmguide	Graphical user interface to examine EPICS medm (*.adl) files
metaio	C library and utilities to read and manipulate LIGO_LW table files
papers	Assorted scripts to help prepare papers for publication
runtools	Summarize status of interferometers during science/engineering runs
segments	Generate and manipulate lists of GPS time intervals
time	GPS/UTC/local time conversion utilities, plus a GPS clock

下图为 ligotools 中的 FrDump 工具，该工具用于查看 LIGO 的数据格式。

```
[root@localhost Desktop]# FrDump -i H-H1_ONLINE_TEST_1-000000000-16.gwf
-----------Parameters used--------------
  Input  Files: H-H1_ONLINE_TEST_1-000000000-16.gwf
  First frame  : 0 0 (GPS=0.0)
  Last  frame  : 2147483647 2147483647 (GPS=2147483647.0)
  Debug level  : 1
 Dump all Frame info
----------------------------------------
H-H1_ONLINE_TEST_1-000000000-16.gwf     0.000000 16  0.000000 0.000000
File(s) summary:
 1 Frames in the requested time range (0 to 10000000000 (GPS))
     First frame start at:0 (UTC:Sat Jan  5 23:59:45 1980) length=16.00s.
        Last frame end at:16 (UTC:Sun Jan  6 00:00:01 1980) length=16.00s.
 ADC :     2 type of AdcData :
        H1:ONLINE-STATE_VECTOR  H1:ONLINE-STRAIN
 Ser :     0 type of SerData :
 Proc:     0 type of ProcData:
 Sim :     0 type of SimData :
 Detector: 0 type of Detector:
 StatData: 0 type of StatData:
 Event   : 0 Types of event in the file
 Simulated Event   : 0 Types of event in the file
```

图1.6　　LIGOtools 中的 FrDump 工具

LSCsoft[6]

LSCsoft 是 LIGO/LSC 中信号分析、数据监测和环境配置的专用软件包集合。该软件包的安装过程如下：

- 提升至 root 权限
- 创建文件/etc/yum.repos.d/lscsoft.repo，输入内容
  [lscsoft]name=LSC Data Analysis Software
  baseurl=https://www.lsc-group.phys.uwm.edu/daswg/download/software/centos/5/$basearch

enabled=1

gpgcheck=0

- 执行 *yum update --disablerepo=lscsoft m2crypto*
- 安装企业版 Linux 扩展包
- 导入 EPEL GPG key，

  执行 *rpm–import/etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL*

使用 yum 安装 LSCsoft 中的包 yum install packagname

表1.2 LSCsoft 中包含的包

Package	Description	Critical Package
FrameL	for data_frame manipulation	libframe, libframe-devel, libframe-utils, libframe-matlab
MetaIO	for LIGO_LW files metadata manipulation	libmetaio, libmetaio-devel, libmetaio-utils, compat-libmetaio
User Environment	LSCSOFT environmental variables definitions	lscsoft-user-env
LAL	LIGO Algorithm Library	lal, lal-devel, lalstochastic, lalstochastic-devel
LALAPPS	LAL based Applications	lalapps
GLUE	Grid LSC User Environment	glue
FrameCPP (deprecated) LDAS-TOOLS	C++ interface to access frame structures	ldas-tools, ldas-tools-general, ldas-tools-general-devel, ldas-tools-genericAPI, ldas-tools-genericAPI-devel, ldas-tools-framecpp, ldas-tools-framecpp-devel, ldas-tools-diskcacheAPI, ldas-tools-diskcacheAPI-devel
DOL	Data Monitoring Tool (DMT) Off_Line	dol
GDS	LIGO Global Diagnostics System	gds-core, gds-crtools, gds-devel, gds-monitors, gds-runtime, gds-web

LSCsoft 中的 DMT (Data Monitor Tools)软件包，承担了 LIGO 数据流监测的功能，DMT Viewer 则是 DMT 中图形化显示数据流状况的程序，如图 1.7 所示。

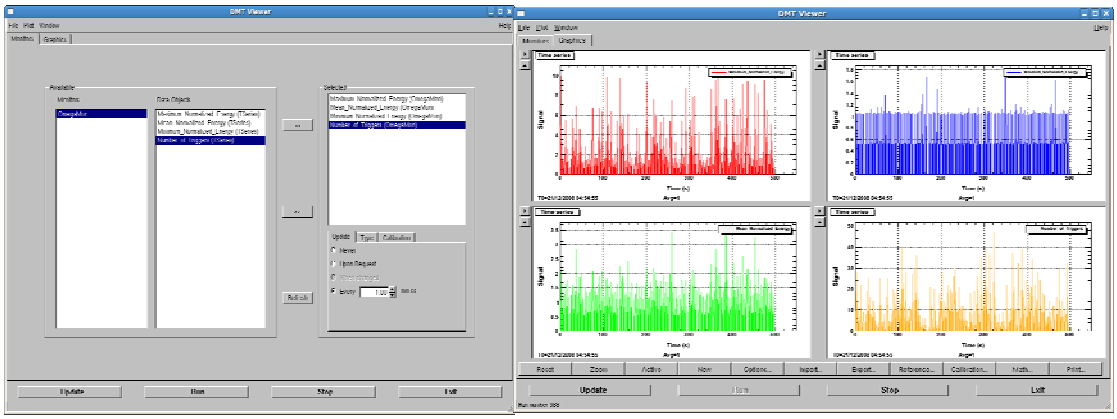

图1.7　　LSCsoft 中的 DMT 工具

LIGO Data Grid[19]

LIGO Data Grid 主要给 LSC 的用户提供以证书的方式登陆集群机，从而无需用户输入密码。LIGO Data Grid 安装过程如下：

- 首先需要配置 LSCsoft 软件仓库，如果没有则执行如下过程
  - 提升至 root 权限
  - 创建文件/etc/yum.repos.d/lscsoft.repo，输入内容

    [lscsoft]name=LSC Data Analysis Software

    baseurl=https://www.lsc-group.phys.uwm.edu/daswg/download/software/centos/5/$basearch

    enabled=1

    gpgcheck=0
  - 执行 yum update --disablerepo=lscsoft m2crypto
- 执行 yum repolist
- 执行 yum install ldg-repo-config
- 通过执行 yum groupinstall ldg-client

1.2 项目目的

LSC 组织为了能更好地管理、处理和监测引力波天文台全天候连续生成的海量数据，专门开发了一系列的数据分析和处理软件。但这些软件均运行在 Linux 环境下，且安装过程较为复杂，部分需要编译后安装，大部分则更需要处理复杂的软件包依赖关系，而 LSC 组织内部的物理学家们往往对此无所适从，在软件的安装和调试上浪费大量的时间和精力。本项目主要目的是根据用户需求制作安装有特定软件组合的虚拟机镜像。此外，根据软件定制化的需求，抽取一般化的软件定制模式，将系统的应用范围扩展到为虚拟机或远程主机自动进行安装、管理软件等操作。

1.3 论文结构

本论文主要介绍了基于虚拟机的 LIGO 软件定制服务系统的实现方法和流程设计，用户在网页上提交 LIGO 软件定制化的需求，系统后台运行一系列虚拟机管理和软件安装脚本实现基于虚拟机的软件定制。

论文第 1 章是引言，介绍了项目产生的背景和主要任务。

论文第 2 章是系统总体设计，给出系统总体设计方案，并对项目用到的几种主要工具进行了介绍。

论文第 3 章是系统实现，介绍了系统的文件结构，程序设计，以及实现效果。

论文第 4 章是总结展望，对项目进行总体总结，并分析了该项目在哪些方面可以进行改进。

1.4 小结

本章介绍了该项目产生的背景，介绍了 LIGO 设备、LSC 组织以及 LIGO 的几款专业软件，对项目的任务和目的进行了描述，最后给出了论文的结构和组织。

第2章 总体设计

2.1 方案选择

2.1.1 Linux 软件管理方式

Linux 是一个非常优秀且免费的操作系统，与 MS－WINDOWS 相比具有可靠、稳定、速度快等优点，拥有根据 UNIX 版本改进来的丰富而又强大功能。Linux 的软件管理与 MS-WINDOWS 也有着巨大的区别，它有 Tar 包，rpm 或 deb 包以及 yum 或 apt 集成管理等几种方式，介绍如下。

使用 Tar 包管理软件

使用 tar 包进行软件安装是最原始也是最通用的方式。Tar 包是一种与 Windows 系统里面的 Zip 文件和 Mac 系统里面的 Sit 类似的压缩文件包。Tar 包通常以后缀名.tar，.tar.gz 或者.tgz 结尾。其安装过程通常为为，下载程序源文件包，自行解压缩，在解压缩出来的文件中找到并阅读 README 或 INSTALL 文件。如果解压缩出来的文件中有可执行脚本，则通常是安装脚本，更通常的情况是解压缩出来的是软件的源文件，这时候在安装之前需要先进行配置和编译。这种软件管理方式比较麻烦，易出错，现在已逐渐被 YUM 和 APT 等更先进的集成软件管理方式取代。但是 Tar 包为一些专业用户提供了自行修改调试的机会。

使用 rpm 或 deb 包管理软件

Red Hat 包管理器（即 RPM：Red Hat Package Manager）最早由 Red Hat 研制，现在也由开源社区开发，用于软件的发布和安装，是 Linux 下广泛使用的包管理器，也是 GNU/Linux 下资源最丰富的软件包类型。RPM 软件包分为二进制包（Binary）、源代码包（Source）和 Delta 包等三种。二进制包以.rpm 为后缀名，可以直接安装在计算机中；而源代码包通常以 src.rpm 为后缀名，将会由 RPM 自动编译、安装。RPM 包的程序的安装、升级和删除等管理由 rpm 程序来处理，而 rpm 程序也仅适用于安装用 RPM 来打包的软件。

DEB 与 RPM 包管理器类似，它是 Debian 软件包格式，文件扩展名为.deb。Deb 包是 Unixar 的标准归档，将包的文件信息以及包内容，经过 gzip 和 tar 打包而成。处理这些包的经典程序是 dpkg，也可以通过 apt 来进行操作。

无论是 RPM 包，还是 DEB 包，一个包是一个压缩文件，其中包含了安装一个应用所需的多个文件。尽管包中包含了安装时所需的文件，但是应用程序为了

能运行可能还需要其他文件或其他未包含包的存在，如特定的库等。这样的需求就是包的依赖性（dependency）。对 RPM 或 DEB 包进行处理时，需要手动处理它们的包依赖性。

使用 YUM 或 APT 管理软件

现在流行的 Linux 发行版有自己的集成软件管理解决方案，将常用的软件集中到一个称为“软件仓库”的服务器上，“软件仓库”提供自动下载，计算依赖性，安装软件等服务，用户通常只需要几个命令就可以完成软件管理。大多数软件仓库一般为使用 Yum 或者 apt-get 进行软件管理。

Yum 是 rpm 系统自动更新和包自动安装/移除工具，它是以 Red Hat Fedora 为核心的 Linux 系统的默认工具，并且在其他的 Linux 系统也可以使用。它自动计算依赖关系并指出安装过程会出现的事件。它使得管理成群的机器变得更容易，不再需要手动地为每一个机器使用 rpm 包进行更新。Yum 还可以通过插件来增加自身特性。

Apt-get 是 Debian 的 Deb 软件包管理工具，它的最低底层还是调用 dpkg 包管理程序，它是 Debian 发行版的特点之一。要使用好 apt-get 就要配置好一个名为 sources.list 的资源列表，资源列表指向 Debian 系统的软件库，apt-get 会从该软件库下载安装各种软件包。通过使用 apt-get 可以方便地进行软件的安装、卸载及其他管理。

Computer Cluster 集群管理

计算集群（computer cluster）[7]是一组连接的计算机，在一起紧密地工作，在很多方面看起来像是一个独立的计算机。一个集群的组件通常通过高速的局域网彼此连接。集群通常用来提高计算机的性能。

集群的种类

- 高可靠性集群（High-availability cluster，HA cluster。也被称为失效备援集群，Failover Cluster）主要用来提高集群提供服务的可靠性。当系统服务失效时，通过增加冗余的节点来提供替代服务。最常用的 HA 集群有两个节点，其中一个是具有最少必要条件用来提供冗余。HA 集群使用冗余的集群组件来剔除失效的单个节点。
- 平衡负载集群（Load-balancing cluster）：平衡负载集群是连接在一起的像一个虚拟计算机一样分担计算负载或执行程序的多个计算机。逻辑上，从用户角度来看，它们是多个计算机，但是功能上却作为一个虚拟计算机。用户的请求被管理，在所有的独立的计算机中分发而形成一个集群。这样导致在不同

的机器上平衡计算量，提升集群性能。

- 计算集群（Compute cluster）。一个计算机集群通常用于大规模计算，而不是处理如 Web 或数据库等以 IO 为目的的服务。例如，一个集群可能支持计算天气预报，一个单独的计算机需要经常与其他节点通信，这意味着集群有一个共享的专用网络，拥有许多密集的本地的甚至是同类的节点。
- 网格计算（Grid computing）。网格计算通过利用大量异构计算机（通常为桌面）的未用资源（CPU 周期和磁盘存储），将其作为嵌入在分布式电信基础设施中的一个虚拟的计算机集群，为解决大规模的计算问题提供了一个模型。网格计算的焦点放在支持跨管理域计算的能力，这使它与传统的计算机集群或传统的分布式计算相区别。网格计算与通常的集群计算二者之间主要的不同就是：集群是同构的，而网格是异构的；网格扩展包括用户桌面机，而集群一般局限于数据中心。网格计算的设计目标是解决对于任何单一的超级计算机来说仍然大得难以解决的问题，并同时保持解决多个较小的问题的灵活性。这样，网格计算就提供了一个多用户环境。它的第二个目标就是：更好的利用可用计算力，迎合大型的计算的断断续续的需求。这里面隐含着使用安全的授权技术，以允许远程用户控制计算资源。

集群（Cluster）是一组网络通信设备的集合，集群管理的主要目的就是解决大量分散的网络设备的集中管理问题。集群管理具有以下优点：

- 节省公网 IP 地址。
- 简化配置管理任务。网络管理员只需在一台设备上配置公网 IP 地址就可实现对集群中所有设备的管理和维护，而无需登录到每台设备上进行配置。
- 提供拓扑发现和显示功能，有助于监视和调试网络。
- 可同时对多台设备进行软件升级和参数配置，且不受网络拓扑和距离限制。

2.1.2 为什么选用虚拟机

虚拟机基于他们与真实机器通信的成为而被分为两种。系统虚拟机（system virtual machine）提供支持整个操作系统运行的完全系统平台。与此相反，进程虚拟机（process virtual machine）用来运行一个单独的程序，这意味着它支持一个单独的进程。虚拟机的一个必备特点为，软件的运行被限制在虚拟机所提供的资源和抽象范围内，它不能突破它的虚拟世界。

系统虚拟机

系统虚拟机有时也叫硬件虚拟机，允许在不同的虚拟机之间分享底层的物理硬件资源，每一个虚拟机运行一个它自己的操作系统。在软件层面上提供的虚拟被称为虚拟机监视器（virtual machine monitor 或 hypervisor）。监控程序可以在硬件上或一个操作系统上运行。

系统虚拟机的主要好处有“

- 在一台机器上可以建立彼此高度隔离的多操作系统环境。
- 虚拟机可以提供与宿主机器不同的指令集架构（ISA）。
- 提供应用程序管理的高可靠性和灾难复原性。

系统虚拟机的主要缺点有：

- 由于虚拟机间接使用硬件，它比真实的机器效率低。

多虚拟机运行它们各自的客户操作系统通常用来服务器整合，由独立的机器来运行不同的服务。这通常被称为“服务质量隔离”。

由于允许对一台计算机的时分复用，运行多操作系统的需求成为发展虚拟机最初的动力。系统虚拟机技术使得在同一台计算机上可以运行不同的客户操作系统。例如，Microsoft Windows，Linux 或 Mac，或者是一种系统的老版本以支持最新版本还不支持的软件。在嵌入式系统领域，使用虚拟机支持不同客户操作系统变得流行；一个典型的应用是在支持实时操作系统的同时运行如 Linux 或 Windows 这样高层的操作系统。另一项应用是把可能是尚在开发中的或不受信任的操作系统装入沙箱，与真实系统隔离开来。虚拟机为操作系统的开发提供了其他的便利条件，如便捷的调试通道和更快的重启动等。

进程虚拟机

进程虚拟机有时也被称为应用虚拟机，在操作系统里面作为一个普通程序来运行，并且支持一个单独的进程。当进程开始的时候，进程虚拟机被创建；进程退出的时候，它被销毁。它的目的是提供平台独立的编程环境，这个环境经过抽象与底层硬件或操作系统的细节分离开，允许一个程序在任何平台已同样的方式运行。

进程虚拟机提供高级编程语言的抽象化。进程虚拟机使用解释器，与编译运行的编程语言相比达到了即时编译，即时执行。这种虚拟机目前随着使用 JAVA 虚拟机的 JAVA 语言的流行而变得流行起来。其他进程虚拟机的例子有，Parrot 虚拟机，它为几种解释型语言提供一个抽象层；.NET Framework 运行在一种叫做“Common Language Runtime”的虚拟机上。

一种特殊的进程虚拟机是由计算机集群通信机制抽象的系统。组成这样一个虚拟机的不是单独的进程，而是集群中的每个物理机器上的进程。它们被设计用来减轻并行计算编程的任务，使得程序员可以将注意力集中到算法上。[8]

选择虚拟机作为软件定制的载体的主要原因基于虚拟机的以下优点：

- 自包含，在虚拟机中把软件安装完成后，向用户提供镜像，用户得到后打开就可以使用，无需另行配置。
- 与平台无关，无论是当前主机是 Windows 还是 Linux，都可以通过虚拟机软件来使用虚拟机镜像中的专业软件。
- 可以自定义软件及版本，很多用户作为相关领域的专家可能会需要自行修改定制其需要的软件。而 Cluster 只能提供特定版本的软件。
- 移植方便，用户之间可以交流镜像文件。

同时虚拟机也有以下缺点：

- 虚拟机镜像文件通常较大。
- 在虚拟机中运行软件会占用更多的系统资源，降低软件的性能。

虚拟机的这些劣势在本系统中不作为主要考虑因素。这些综合考察本系统的需求和虚拟机的特点，决定利用虚拟机的优势，采用虚拟机作为软件定制服务的载体。

2.2 需求分析

本系统中，用户访问系统主机的 web 页面，定制虚拟机；本地脚本检测到有新的任务到来，开始执行虚拟机定制脚本。分析系统的功能需求，该系统功能分为两个部分，即：

- Web 网页功能
  - 选择 Linux 发行版、版本号及操作系统位数。
  - 设置要创建的普通用户名和密码。
  - 选择要安装的软件，设置 ligotools 安装位置，要安装的选择 lscsoft 软件包。选择是否安装 ligo data grid。
  - 留下 Email 联系方式，虚拟机定制完成后将下载链接发送给用户，用户自行下载。
- 虚拟机定制系统软件安装功能
  - 由于该系统面向的对象为 LSC 组织里的物理学家们，属于特定的少

数人群，所以将任务检查周期设置为 15 分钟。后台监控进程每十五分钟检查一下是否有新的任务需要执行，如果有则执行下面流程。

- 设置系统繁忙标志位，防止同时运行多个虚拟机。
- 启动相应虚拟机。
- 登陆到虚拟机中。
- 执行安装软件命令，自动回答软件安装程序所提出的问题。
- 退出登录，关闭虚拟机。
- 导出虚拟机，生成下载链接。
- 系统繁忙标志位置零。
- 向用户发送邮件，提供下载链接。

- 虚拟机定制系统系统维护功能
  - 定期清理过期的镜像文件。
  - 使用情况统计，包含使用人数统计，使用趋势统计，选择软件统计等。

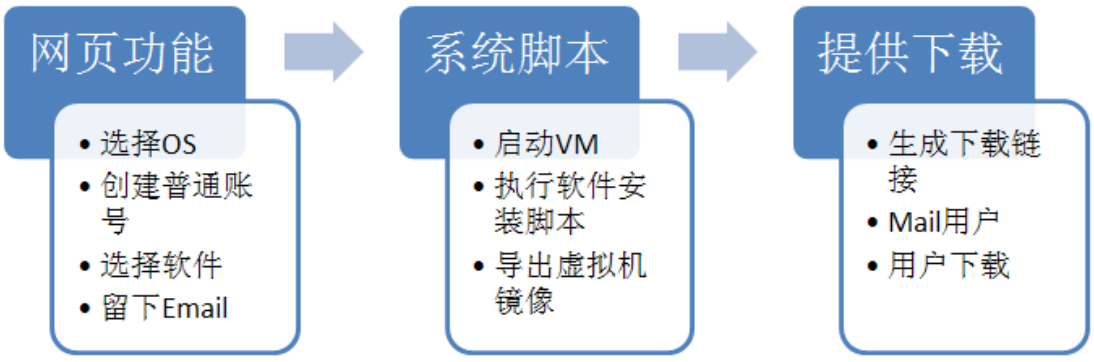

图2.1 系统主要功能

2.3 工具介绍

基于上述需求分析，本系统使用 Apache 构建 Web 服务器；使用 HTML 编写 Web 页面；使用 Shell 编写系统脚本；使用 Expect 编写软件安装脚本。

表2.1 系统需求与所用工具

需求	服务器	Web 页面	系统运行脚本	软件安装脚本
工具	Apache	HTML+PHP	Shell	Expect

2.3.1 PHP

PHP[14][15][16]是一种广泛使用的通用脚本语言，特别适用于 Web 开发，并且可以嵌入在 HTML 中[13]。PHP 最初代表“个人主页”（Personal Home Page），由 Rasmus Lerdorf 于 1994 年创建，用于跟踪访问者对其在线履历的访问。随着实用性和功能性的不断提高（并且也开始用于更专业的环境中），它变成了“PHP：Hypertext Preprocessor（PHP 超文本预处理器）”。PHP 可以嵌入到 HTML 中就意味着，在标准的 HTML 页面中根据需要插入一些 PHP 代码，就可以得到动态效果。因此 PHP 适合网页设计和制作者使用。

2.3.2 Shell

Linux 提供了大量的命令，利用它可以有效地完成大量的工作，如磁盘操作、文件存取、目录操作、进程管理、文件权限设定等。所以，在 Linux 系统上工作离不开使用系统提供的命令。在 Linux 操作系统系，用户通常通过直接输入命令来执行任务。Linux 下的图形用户界面 GNOME 和 KDE，有时也被叫做虚拟 Shell 或者图形 Shell。Shell 不仅为用户提供交互的界面，而且许多控制系统的脚本也是由 Shell 写成的，例如在 Linux 系列操作系统下，Shell 控制着系统启动，X Window 启动等重要系统功能。Shell 可以实现简单的控制流功能，如循环、判断等，Linux 用户可以使用 Shell 脚本将 Linux 命令进行组合，进而形成强大的功能[17][18]。

2.3.3 Expect

Expect[10][12]是一个为如 telnet，ftp，passwd，fsck，rlogin，tip 等应用提供自动交互的工具。通过增加 Tk，Expect 可以与 X11 的图形界面进行交互。Expect 含有利用正则表达式进行模式匹配以及通用的编程功能，允许简单的脚本智能地管理如下工具：telent，ftp 和 ssh（这些工具都缺少编程的功能），宏以及其它程序。Expect 脚本的出现使得这些老的软件工具有了新的功能和更多的灵活性。在本系统中，使用 Expect 实现与虚拟机通信，并在虚拟机中执行软件安装命令。

2.3.4 Virtualbox

Virtualbox[12]是一个跨平台的虚拟化软件。它可以安装在基于 Intel 或 AMD 的计算机上，无论它们是运行着的是 Windows，Mac，Linux 还是 Solaris 操作系统。Virtualbox 的主要特点如下：

- **模块性。**Virtualbox 具有内部良好定义的模块化设计和客户端/服务器模

式设计。这使得可以一次轻易地控制几个接口，例如，可以在典型的虚拟机 GUI 中启动虚拟机，然后在命令行中或远程控制这个虚拟机。Virtualbox 带有一个完整的软件开发工具包，它甚至一个开源软件。

- **以 XML 描述虚拟机。**虚拟机的配置信息完全存储在 XML，并且完全本地机器相隔离。虚拟机的定义等信息可以轻易地转移到其他机器上。
- **提供 Guest Additions。**Virtualbox 有专门的软件用来安装在 Windows，Linux 和 Solaris 虚拟机内部来提升性能，并使虚拟机与宿主机实现无缝集成。在 Guest Additions 提供的特性中包含集成鼠标，任意改变屏幕分辨率在内的主要功能有：
  - 使用 host 机器上共享的硬盘，使用方法：mount -t vboxsf<共享名>< 本地目录>。
  - 鼠标可以自由出入 vbox 窗口。
  - 自动与 host 同步时间。
  - 自动根据窗口大小 Virtualbox 改变 X 尺寸。
  - 与 host 共享剪贴板。
- **共享文件夹。**像其他虚拟化解决方案一样，Virtualbox 提供主机与客户机之间的数据交换。Virtualbox 允许定义主机中特定的文件夹为“share folders”，虚拟机可以加载并访问这些文件夹。

2.3.5 Crontab 命令

在 Linux 系统中，crontab 是用于设置周期性执行脚本的命令。该命令将参数和指令存放于 crontab 文件中，后台运行的 crond 检查是否有预定的作业需要执行。。每个用户拥有自己的 crontab 文件，而操作系统保存着整个系统的 crontab 文件。

Crontab 命令格式如下：

crontab [-e [UserName]|-l [UserName]|-r [UserName]|-v [UserName]|File]

参数:

-e [UserName]: 执行文字编辑器来设定时程表。

-r [UserName]: 删除目前的时程表。

-l [UserName]: 列出目前的时程表。

-v [UserName]:列出用户 cron 作业的状态。

使用方法:

编辑一个文件 cronfile，然后在这个文件中输入正确格式的时程表。编辑完成后，保存并退出。在命令行输入 crontab cronfile。

Crontab 文件的格式如下：

crontab 文件的每一行均遵守特定的格式，由空格或 tab 分隔为数个领域，每个领域可以放置单一或多个数值。其命令格式为：Crontab 文件每行有六个参数。第一个参数为分，第二个参数为时，第三个参数为天，第四个参数为月，第五个参数为周，第六个参数为要周期执行的命令及参数。各参数之间以空格分界，每个参数内可以设置多个值，其间以分号为界。

如图 2.2 所示。

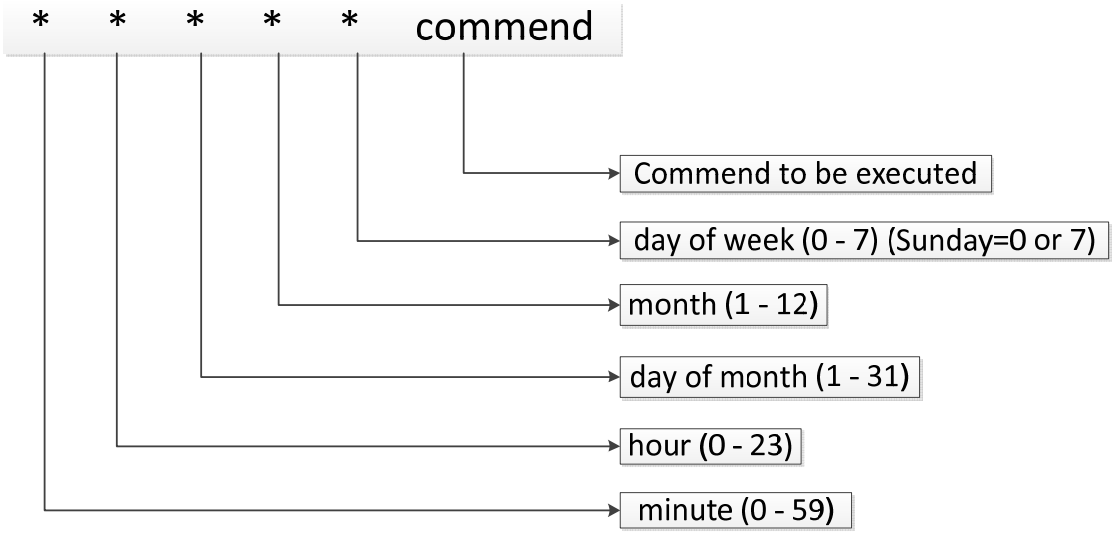

图2.2 crontab 文件格式示意图

2.4 小结

本章分析并比较了 Linux 下多种软件管理方式，给出选择虚拟机作为软件管理载体的理由。对项目需求进行了详细的分析，并介绍了系统搭建用到的 PHP，Shell，Expect，Virtualbox 等工具以及 crontab 命令。

第3章 系统实现

3.1 系统准备

3.1.1 系统软件及服务准备

本系统具有通用性，可配置在任意 Linux 发行版下，但在本文中以 Ubuntu 作为系统实现环境。Ubuntu 是一个由开源社区开发的，适用于笔记本电脑、桌面电脑和服务器。无论是在家庭、学校还是工作时使用，Ubuntu 都包含了无论是文字处理和电子邮件，还是 Web 服务和编程工具等所有必须的工具。Ubuntu 是现在最为流行的 Linux 发行版之一。

构建本系统，需要完成以下几步系统准备。

- 安装 Virtualbox，使用 Virtualbox 作为虚拟机软件。
- 安装 ssh 服务，使系统可支持远程登录。
- 安装 tcl 和 expect，使得系统可以支持 expect 脚本编程。
- 安装 Apache 和 PHP，启动 Apache 服务，使得系统可以支持 Web 访问。
- 安装 MySQL 数据库，安装 MySQL 与 PHP 关联软件。
- 安装 mail 服务器，使得系统支持发送邮件服务。

本系统提供一个系统准备脚本，这个脚本所在的文件夹里包含有为构建该系统所准备的软件工具，所需的 Web 页面文件，所有系统运行和维护脚本，其内容为：

（1） 文件夹 ligo，其中包含有系统运行和维护的所有脚本。

（2） 文件夹 www，其中包含有 Web 服务器的页面显示所需的所有页面文件。

（3） 文件 tcl8.4.19-src.tar.gz，为 tcl 的源程序包，需要编译安装。

（4） 文件 expect.tar.gz，为 expect 的 tar 程序包，需要编译安装，在安装前需要先安装 tcl，还需要引用 tcl 的头文件。

（5） 脚本 install-sh，为系统准备脚本，其功能为进行系统配置如安装 ssh 服务，Apache 服务，安装 PHP，mail 服务等，解压 tcl 和 expect 两个压缩文件，并分别对其编译安装，将文件 ligo 内的脚本文件和文件夹 www 内的页面文件拷贝到指定位置，设置 crontab 文件等等操作，执行完 install-sh 脚本后，该系统即可成功运行。

系统准备主要流程如图 3.1 所示。

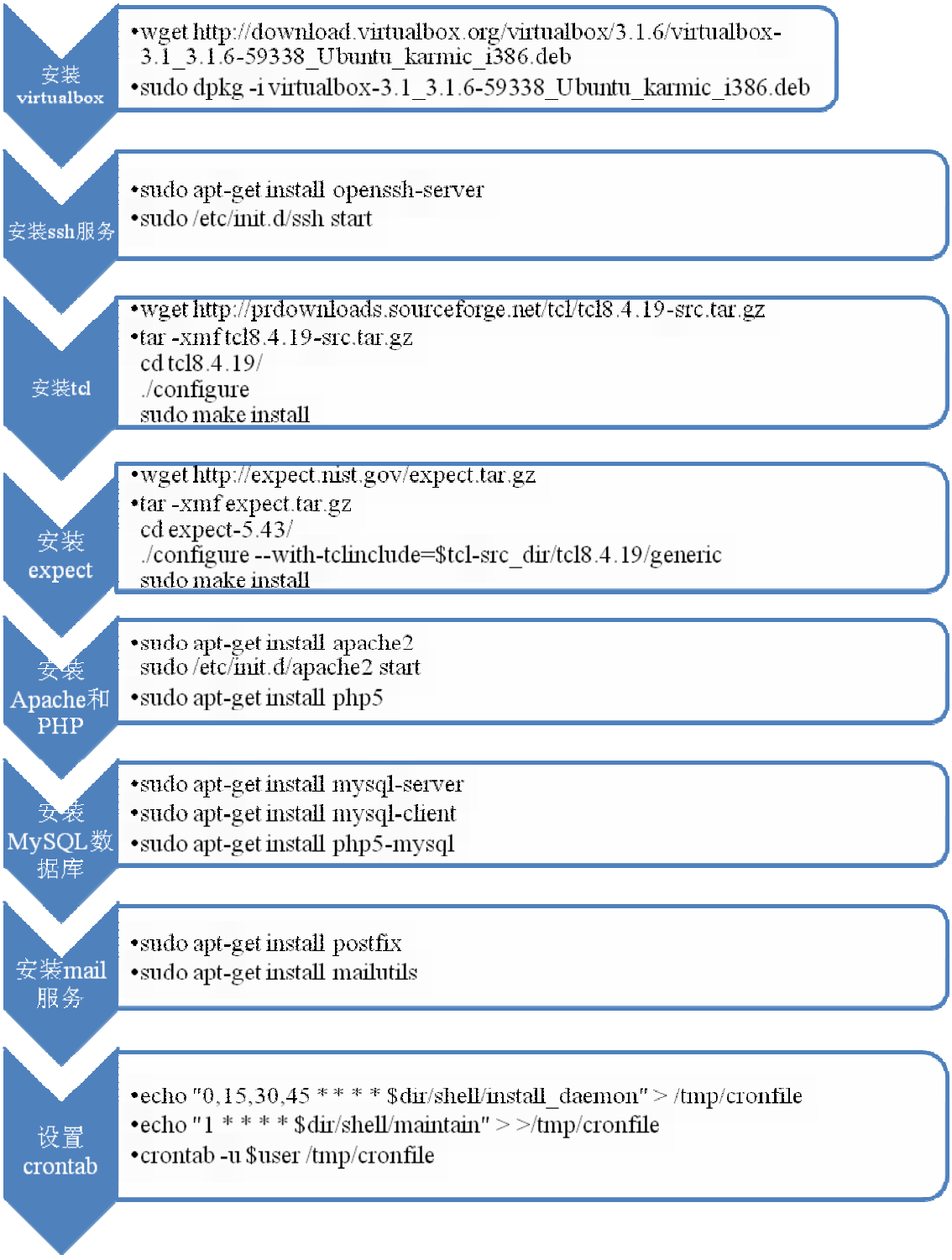
安装 virtualbox
•wget http://download.virtualbox.org/virtualbox/3.1.6/virtualbox-3.1_3.1.6-59338_Ubuntu_karmic_i386.deb
•sudo dpkg -i virtualbox-3.1_3.1.6-59338_Ubuntu_karmic_i386.deb
安装ssh服务
•sudo apt-get install openssh-server
•sudo /etc/init.d/ssh start
安装tcl
•wget http://prdownloads.sourceforge.net/tcl/tcl8.4.19-src.tar.gz
•tar -xmf tcl8.4.19-src.tar.gz
cd tcl8.4.19/
./configure
sudo make install
安装 expect
•wget http://expect.nist.gov/expect.tar.gz
•tar -xmf expect.tar.gz
cd expect-5.43/
./configure --with-tclinclude=$tcl-src_dir/tcl8.4.19/generic
sudo make install
安装 Apache和 PHP
•sudo apt-get install apache2
sudo /etc/init.d/apache2 start
•sudo apt-get install php5
安装 MySQL数据库
•sudo apt-get install mysql-server
•sudo apt-get install mysql-client
•sudo apt-get install php5-mysql
安装mail 服务
•sudo apt-get install postfix
•sudo apt-get install mailutils
设置 crontab
•echo "0,15,30,45 * * * * $dir/shell/install_daemon" > /tmp/cronfile
•echo "1 * * * * $dir/shell/maintain" > >/tmp/cronfile
•crontab -u $user /tmp/cronfile

图3.1 系统准备图

3.1.2 数据库准备

安装好 MySQL 数据库服务之后，创建数据库 ligoinfo，在数据库 ligoinfo 中创建 info，ligotools，lscsoft，ldg 等表。各表之间以任务的序列号为关联关系。数据库中的表如图 3.2 所示。

```
mysql> show tables;
+--------------------+
| Tables_in_ligoinfo |
+--------------------+
| info               |
| ldg                |
| ligotools          |
| lscsoft            |
+--------------------+
mysql> describe info;
+-----------+------------------+------+-----+---------+----------------+
| Field     | Type             | Null | Key | Default | Extra          |
+-----------+------------------+------+-----+---------+----------------+
| user_id   | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| task_id   | int(10) unsigned | NO   |     | NULL    |                |
| email     | varchar(40)      | NO   |     | NULL    |                |
| os        | varchar(20)      | NO   |     | NULL    |                |
| bit       | char(5)          | NO   |     | NULL    |                |
| ligotools | tinyint(4)       | NO   |     | NULL    |                |
| lscsoft   | tinyint(4)       | NO   |     | NULL    |                |
| ldg       | tinyint(4)       | NO   |     | NULL    |                |
| task_date | datetime         | NO   |     | NULL    |                |
+-----------+------------------+------+-----+---------+----------------+
mysql> describe ldg;
+---------+------------------+------+-----+---------+----------------+
| Field   | Type             | Null | Key | Default | Extra          |
+---------+------------------+------+-----+---------+----------------+
| ldg_id  | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| user_id | int(10) unsigned | NO   |     | NULL    |                |
+---------+------------------+------+-----+---------+----------------+
mysql> describe ligotools;
+--------------------+------------------+------+-----+---------+----------------+
| Field              | Type             | Null | Key | Default | Extra          |
+--------------------+------------------+------+-----+---------+----------------+
| ligotools_id       | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| user_id            | int(10) unsigned | NO   |     | NULL    |                |
| ligotools_location | varchar(40)      | NO   |     | NULL    |                |
+--------------------+------------------+------+-----+---------+----------------+
mysql> describe lscsoft;
+------------------------------+------------------+------+-----+---------+----------------+
| Field                        | Type             | Null | Key | Default | Extra          |
+------------------------------+------------------+------+-----+---------+----------------+
| lscsoft_id                   | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| user_id                      | int(10) unsigned | NO   |     | NULL    |                |
| ImageMagick                  | int(10) unsigned | NO   |     | NULL    |                |
| R                            | int(10) unsigned | NO   |     | NULL    |                |
                                .
                                .
                                .
| root                         | int(10) unsigned | NO   |     | NULL    |                |
| root_debuginfo               | int(10) unsigned | NO   |     | NULL    |                |
| scipy                        | int(10) unsigned | NO   |     | NULL    |                |
| spr                          | int(10) unsigned | NO   |     | NULL    |                |
| spr_devel                    | int(10) unsigned | NO   |     | NULL    |                |
| xorg_x11_server_Xvfb         | int(10) unsigned | NO   |     | NULL    |                |
+------------------------------+------------------+------+-----+---------+----------------+
```

（lscsoft 中各软件包）

图3.2 数据库中的表

3.2 系统文件结构

3.2.1 Web 服务器文件结构

Web 页面存储在文件夹/var/www 中，其文件结构为：

```
www
|-- customize.php
|-- downloads
|     |-- 1
|     |     |-- CentOS5.4.mf
|     |     |-- CentOS5.4.ovf
|     |     `-- CentOS5.4.vmdk
|           .
|           .
|     `-- 5
|           |-- CentOS5.4.mf
|           |-- CentOS5.4.ovf
|           `-- CentOS5.4.vmdk
|-- includes
|     |-- config.inc.php
|     |-- footer.html
|     |-- header.html
|     `-- layout.css
|-- index.html
|-- index.php
|-- mysql_connect.php
`-- vms
      |-- 1.php
      .
      .
      `-- 5.php
```

图3.3 /vaw/www 目录结构图

其中各目录及文件作用为：

- 文件夹 includes 包含有页头 header.html 和页脚 footer.html，而 layout.css 为页面的页面布局文件。
- 文件夹 downloads 存储着为用户提供下载的虚拟机镜像文件。该文件夹用普通用户读取，其所属为普通用户。
- 文件夹 vms 内为为用户提供下载链接的页面文件，其文件名为任务序列号。
- 文件 index.html 为服务器的起始页面。
- 文件 customize.php 为用户自定义安装软件的页面，是本系统的主要页面。它完成的功能主要有接受用户输入的软件定制信息，进行格式检查，将用户的软件定制信息写入数据库，并生成任务文档。
- 文件 mysql_connect.php 用于与数据库建立连接，使得 PHP 程序可以对数据库进行读写操作。

3.2.2 软件安装系统文件结构

系统文件内容及作用为：

- 文件 busy 中保存着 0 或 1，为当前系统的工作状态的标志位，若 busy 文件中的值为 0，则说明当前系统中没有正在被执行的任务，否则意味着系统繁忙中。
- 文件 installed.id 中保存着一个整数值，是当前已完成任务的数量，它指示出任务完成进度情况，同时预示着下一个要执行的任务（即当前 installed.id 值加 1）。
- 文件 toinstall.id 中叶保存着一个整数值，是现有的任务的总数。当 toinstall.id 文件的值大于 installed.id 的值时预示着有新的任务到来。当 toinstall.id 的值等于 installed.id 的值时，当前任务全部执行完成。
- 文件夹 shell，其中包含有软件安装所必须的脚本，如 install_daemon，startvm，ligotools，lscsoft，ldg，adduser，closevm，maintain，sendmail 等脚本。3.3.2，3.3.3 和 3.3.4 节将将具体介绍这些脚本内容和流程。
- 文件夹 task，其中包含着根据用户在网页上动态定制的软件安装任务。其中文件是由 PHP 代码动态生成的。文件名如 1、2 就是任务的 id 号，而文件的内容为带参数的软件安装脚本组合而成。

系统文件结构如下图所示。

```
ligo
|-- busy
|-- installed.id
|-- log.txt
|-- shell
|     |-- adduser
|     |-- backup
|     |-- closevm
|     |-- downloadlink
|     |-- install_daemon
|     |-- ldg
|     |-- ligotools
|     |-- lscsoft
|     |-- maintain
|     `-- startvm
|-- task
|     |-- 1
      .
|     `-- 5
`-- toinstall.id
```

图3.4 /ligo 目录结构图

3.2.3 文件权限设置

Linux 系统内对文件有着严格的权限管理制度，各文件的权限分为读，写，可执行等三种。而各文件有其所属的用户和用户组，在没有给定权限的情况下，一个用户是不能访问属于其他用户的文件的。

/var/www 目录下的权限设置：

```
-rw-r--r--    1 root         root         8943 2010-06-09 11:24 customize.php
drwxr-xr-x    6 liziyang liziyang 4096 2010-06-12 10:45 downloads
```

```
drwxr-xr-x  2 root        root        4096 2010-05-17 16:14 includes
-rw-r--r--  1 root        root         973 2010-05-15 14:45 index.html
-rw-r--r--  1 root        root         868 2010-05-15 14:35 index.php
drwxr-xr-x  2 liziyang liziyang 4096 2010-06-12 10:52 vms
```

目录 downloads 和 vms 及两个目录下的文件，为用户所读写，其所属为普通用户。其余文件夹和文件均为 root 所属。

/ligo 目录下权限设置如下所示：

```
-rw-r--r--  1 liziyang liziyang    2 2010-06-13 09:45 busy
-rw-r--r--  1 liziyang liziyang    2 2010-06-12 10:27 installed.id
-rw-r--r--  1 liziyang liziyang  319 2010-06-12 10:59 log.txt
drwxr-xr-x  3 www-data www-data 4096 2010-06-12 10:54 shell
drwxr-xr-x  2 www-data www-data 4096 2010-06-12 10:26 task
-rw-r--r--  1 www-data www-data    1 2010-06-12 10:26 toinstall.id
```

文件 busy，installed.id，log.txt 为用户脚本读写，其所属为普通用户；而文件夹 task 和文件 toinstall.id 由 Web 页面 PHP 程序读写，其所属用户为 www-data。

3.3 系统程序设计

系统程序设计分为三个部分，一为 PHP 页面编程，二为 Expect 交互式脚本编程，三为 Shell 脚本编程。其中：

- PHP 页面编程的作用是在前端 Web 服务器接受用户定制软件信息，将从用户得来的软件定制信息转化为任务脚本，设为可执行模式，供系统脚本读取和执行。
- Expect 交互式脚本编程的作用是与虚拟机内客户操作系统进行通信，实现自动安装软件。
- Shell 脚本为系统内进行总体组织和运行的脚本。

下面介绍各脚本的功能和详细流程。

3.3.1 PHP 页面编程

页面 Customize Page 为用户自定义安装软件的页面。在该页面，用户可以根据需要，填写定制软件信息。用户提交软件定制信息后，页面检查用户输入的正确性，检查完成后，将用户的软件定制信息存入数据库，并生成用户任务脚本，显示用户定制信息。页面文件流程如图 3.5 所示。

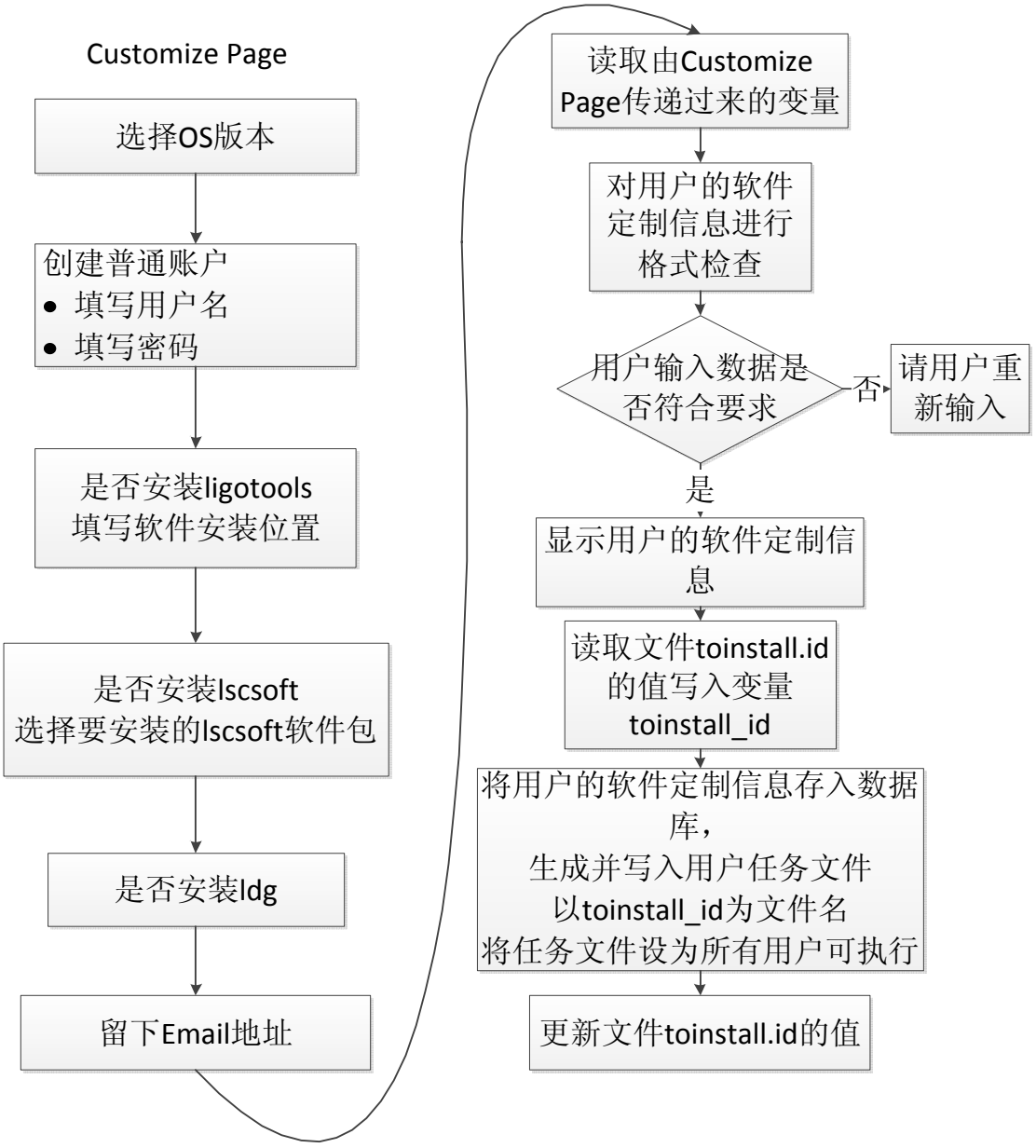

图3.5 Web 页面流程

3.3.2 软件安装脚本——Expect 脚本编程

下面以自动登录过程的脚本为例介绍 Expect 的基本编程方法：

```
#!/usr/local/bin/expect -f
set user root
set password ligosoft
```

#使用 set 设置变量值。

```
set ip 127.0.0.1
set port 2332

spawn ssh -l $user -p $port $ip
```

#spawn 命令通过程序或远程设备建立一个 Expect 可以管理的会话。

```
while ($done) {
```

#通过 spawn 打开一个会话后，可以使用 expect 命令来“期待”会话中可能出现的字符。

#根据“期待”得到的不同字符发送不同的回应，以实现对话。

#send 命令可以向会话中发送字符，其中的“\r”为回车符。

#，如下面的 expect 得到“Are you sure you want to continue connecting (yes/no)?”，#发送“yes\r”；expect 得到“password”，则发送“$password\r”；而 expect 得到“#”则说明登录已成功，退出循环。

```
expect {
    "Are you sure you want to continue connecting (yes/no)?" {send "yes\r"}
    "password:" { send "$password\r" }
    "#" {
        set done 0
        send_user "Login Successfully...\r"
        break
        }
        timeout {
            switch -- $timeout_case {
                0 { send "" }
                1 {
                    send_user "Send a return...\r"
    }
                2 {
                    puts stderr "Login time out...\r"
    exit 1
                }
            }
        incr timeout_case
    }
}
```

脚本 ligotools 为 ligotools 软件的安装脚本，ligotools 软件安装过程如前所示。其参数为 ligotools 安装的位置，例如 ligotools　/root/ligotools。脚本 ligotools 流程如图 3.6 所示。

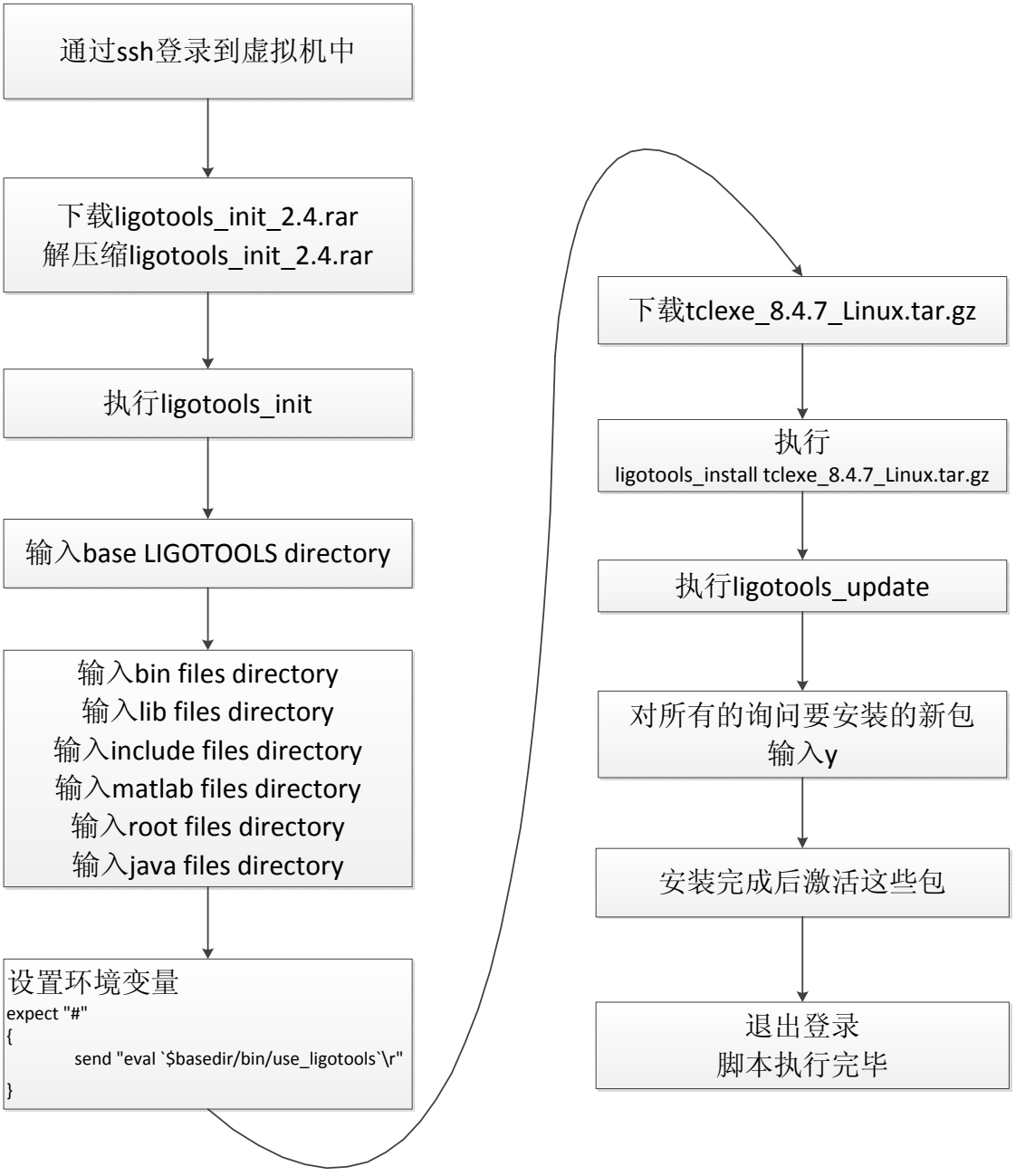

图3.6 ligotools 脚本流程图

脚本 lscsoft 为 lscsoft 软件的安装脚本。其后参数为 lscsoft 中包含的软件包的名称，脚本 lscsoft 依次读取这些参数并安装这些包，其流程如图 3.7 所示。

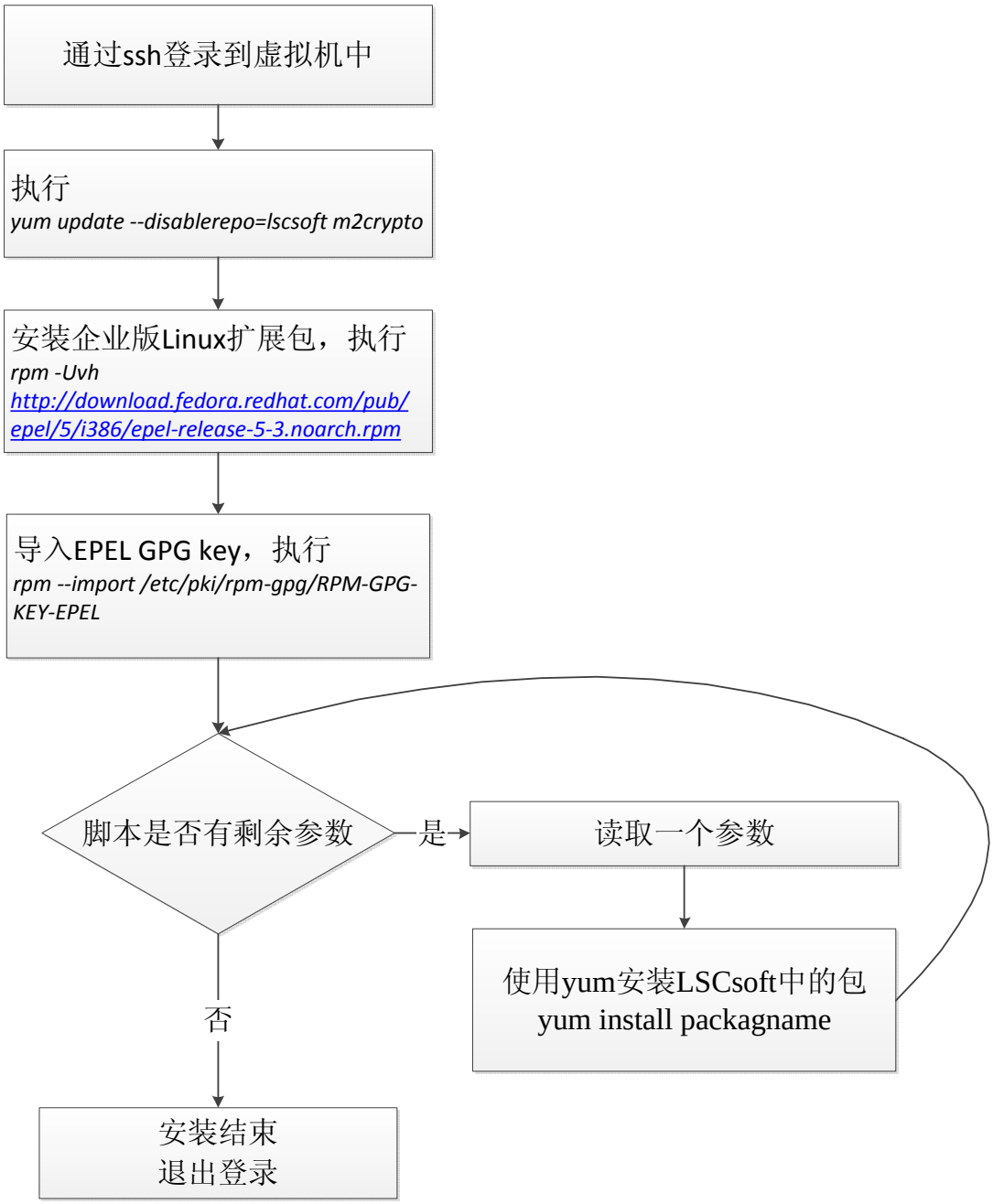

图3.7 lscsoft 脚本流程图

脚本 ldg 为 ldg 软件的安装脚本，流程如图 3.8 所示。

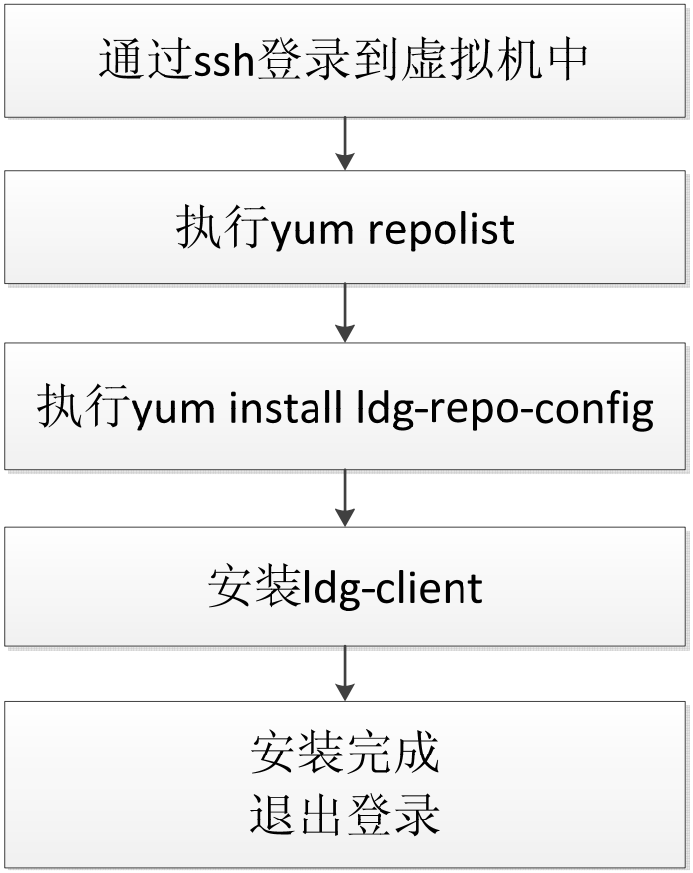

图3.8 ldg 脚本流程图

3.3.3 系统运行脚本

crontab 后台自动周期运行

crontab 文件如下所示：

0,15,30,45 * * * * /ligo/shell/install_daemon

每个小时的第 0，15，30，45 分，即每隔 15 分钟运行脚本 install_daemon。该 install_daemon 脚本为自动周期后台运行。

脚本 install_daemon

脚本 install_daemon 流程图如图 3.9 所示。

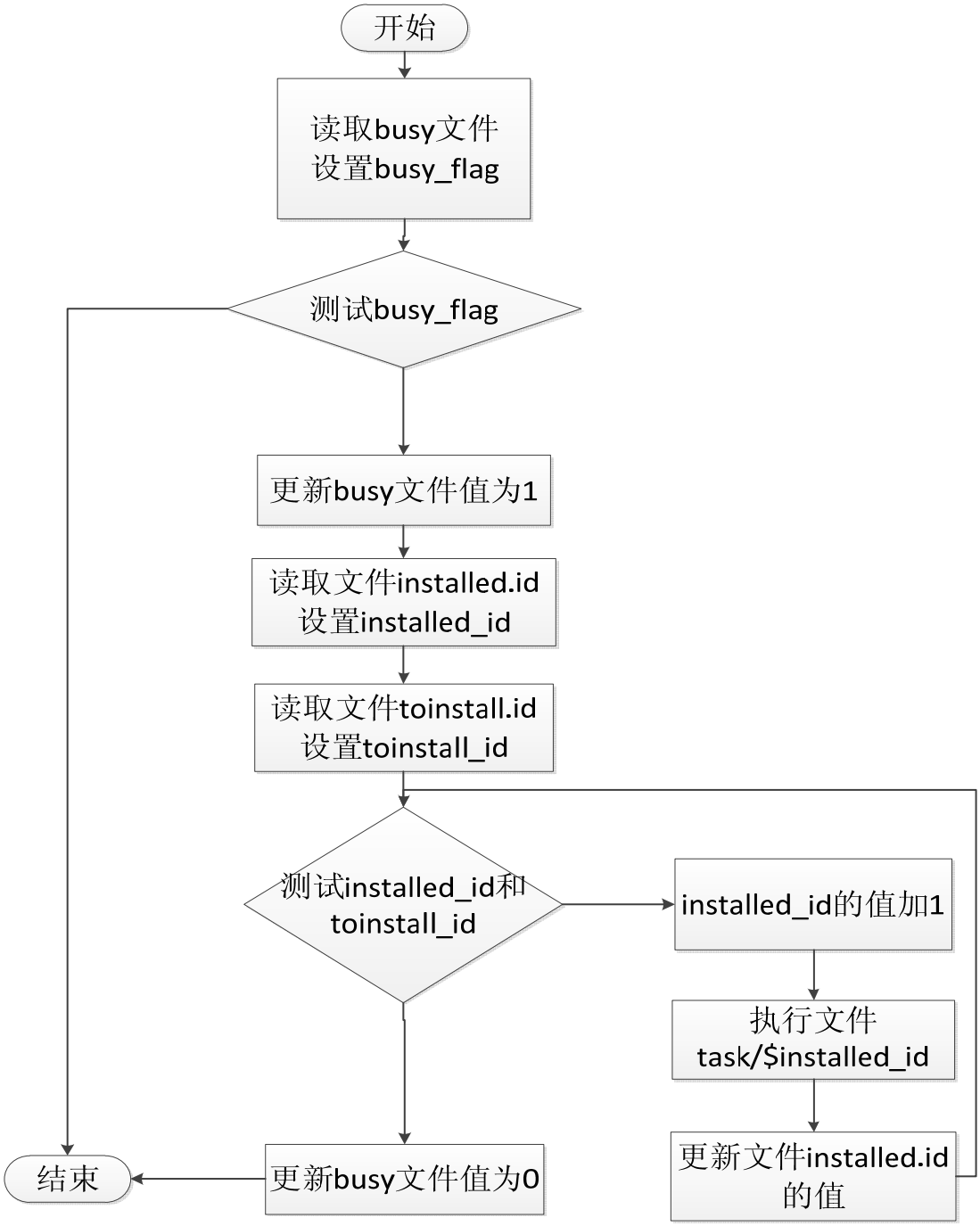

图3.9 脚本 install_daemon 流程图

脚本 install_daemon 的作用是，每隔十五分钟检测一次当前是否有任务在执行及是否有新的任务到来。如果 busy 文件的值等于 0，则说明当前系统空闲，没有任务在执行，否则说明有任务在执行，install_daemon 退出。如果系统空闲，且检测到有新的任务到来，即 installed.id 的值小于 toinstall.id 的值。将 install.id 的值

加 1，执行 task 目录下的第 installed.id 个任务，循环执行这个过程，直到 install.id 等于 toinstall.id 为止。当所有任务执行完成后，即 installedid 等于 toinstall.id 时，将 busy 文件内容置为零，install_daemon 退出。

脚本 startvm

脚本 startvm 的作用是通过使用 VBoxManage 中的 setextradata 命令，将主机的一个端口（如 2332）与虚拟客户系统的端口 22（即 ssh 默认使用的端口）建立映射关系。这样访问主机的端口 2332 即相当于访问端口 22。然后，使用 VBoxManage 中的 startvm 命令启动虚拟机，加参数—type headless 可不显示虚拟客户系统的 GUI。最后 sleep 250，即暂停 250 秒，等待虚拟系统启动完成。其流程如图 3.10 所示。

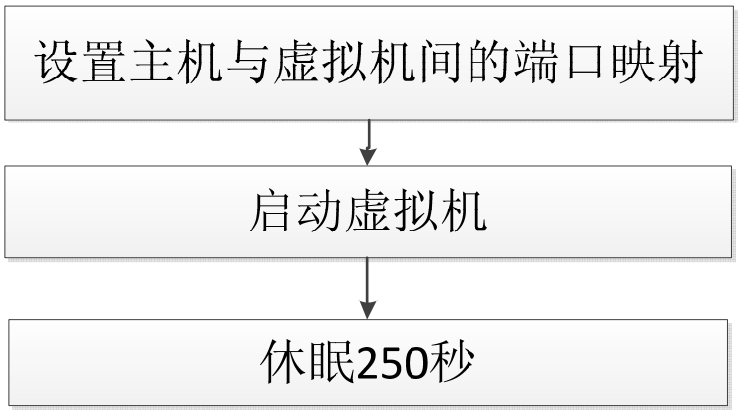

图3.10 脚本 startvm 流程图

代码如下所示：

```
#!/bin/dash
os=$1
echo "$os"
VBoxManage setextradata "$os"
"VBoxInternal/Devices/pcnet/0/LUN#0/Config/guestssh/Protocol" TCP
VBoxManage setextradata "$os"
"VBoxInternal/Devices/pcnet/0/LUN#0/Config/guestssh/GuestPort" 22
VBoxManage setextradata "$os"
"VBoxInternal/Devices/pcnet/0/LUN#0/Config/guestssh/HostPort" 2332
VBoxManage startvm $os --type headless
```

sleep 250

脚本 closevm

脚本 closevm 的作用是关闭虚拟机，导出虚拟机，生成下载链接。其流程如图 3.11 所示。

图3.11 脚本 closevm 流程

任务脚本流程

任务脚本为页面 confirm.php 根据用户定制的软件信息自动生成的，其实际由用户 www-data 写入，其包含主要内容有 startvm，adduser，ligotools，lscsoft，ldg，closevm，backup，发送邮件等脚本。如下以任务 5 为例，任务脚本的流程图如图 3.12 所示。

图3.12 任务脚本流程图

3.3.4 系统维护脚本

系统维护脚本 maintain 的主要作用为定期清理已经过期的虚拟机镜像文件。脚本读取镜像文件的修改时间与当前系统时间相比，如果该镜像文件已存在超过一定时间（如 7 天），则将该镜像文件删除。

3.3.5 系统总体流程

系统总体流程如图 3.13 所示。

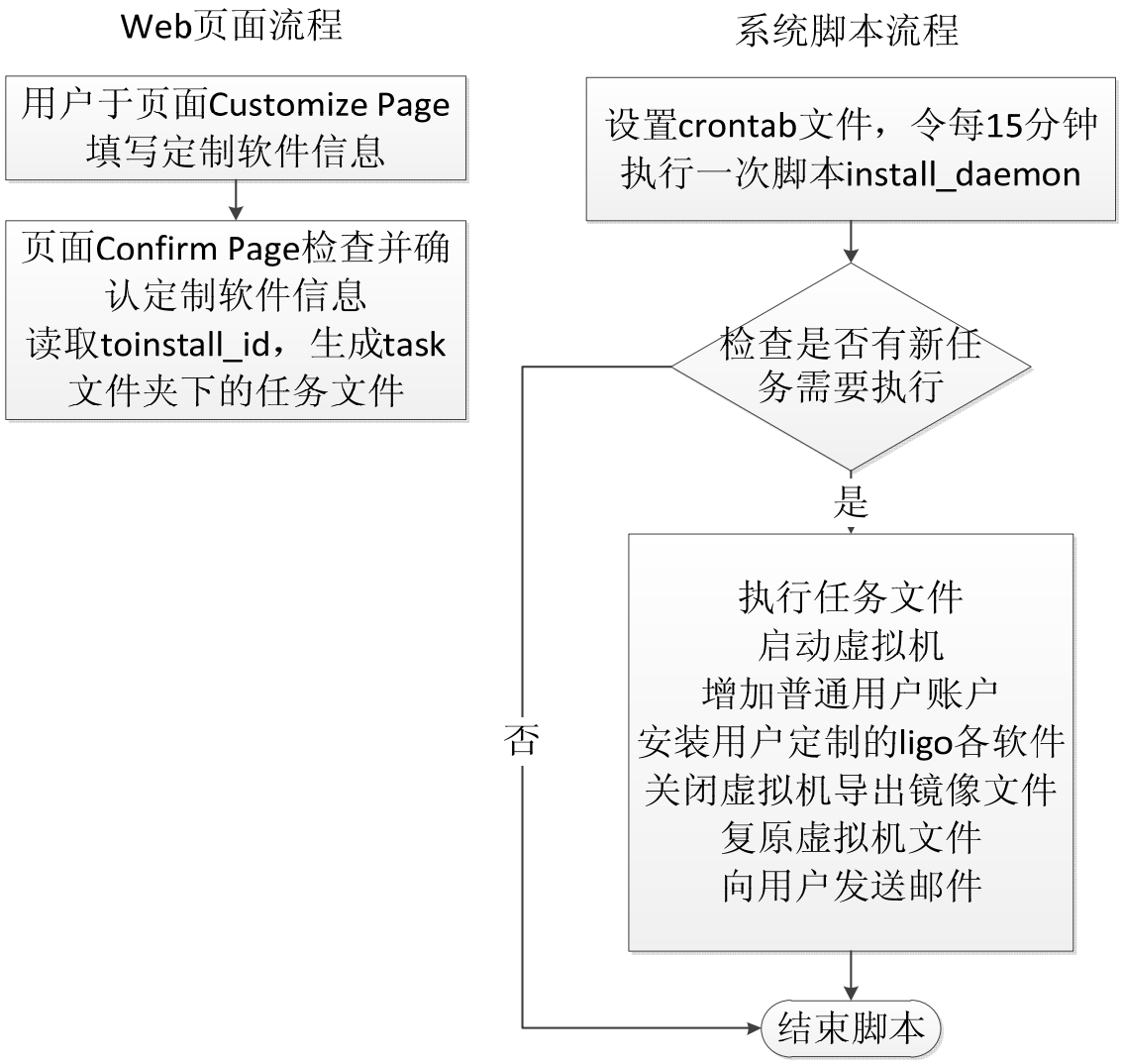

图3.13 系统总体流程图

3.4 系统效果

系统运行脚本和系统维护脚本都是后台运行的，在实际系统中均无截图。下面为基于虚拟机的 LIGO 软件定制系统的 Web 页面截图。

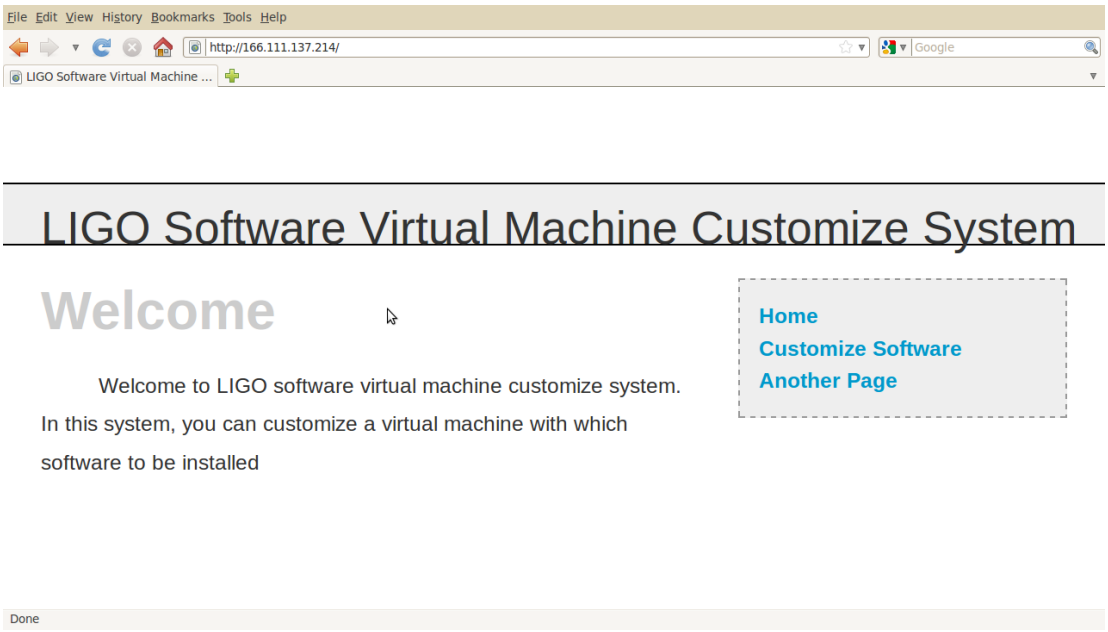

图3.14 起始页面

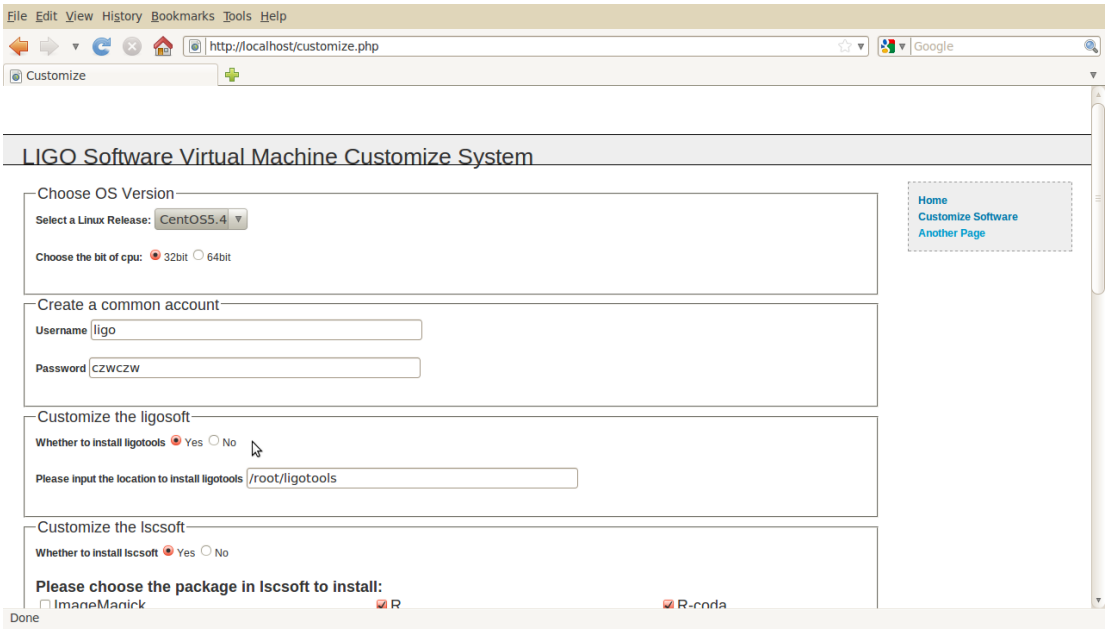

图3.15 定制页面（1）

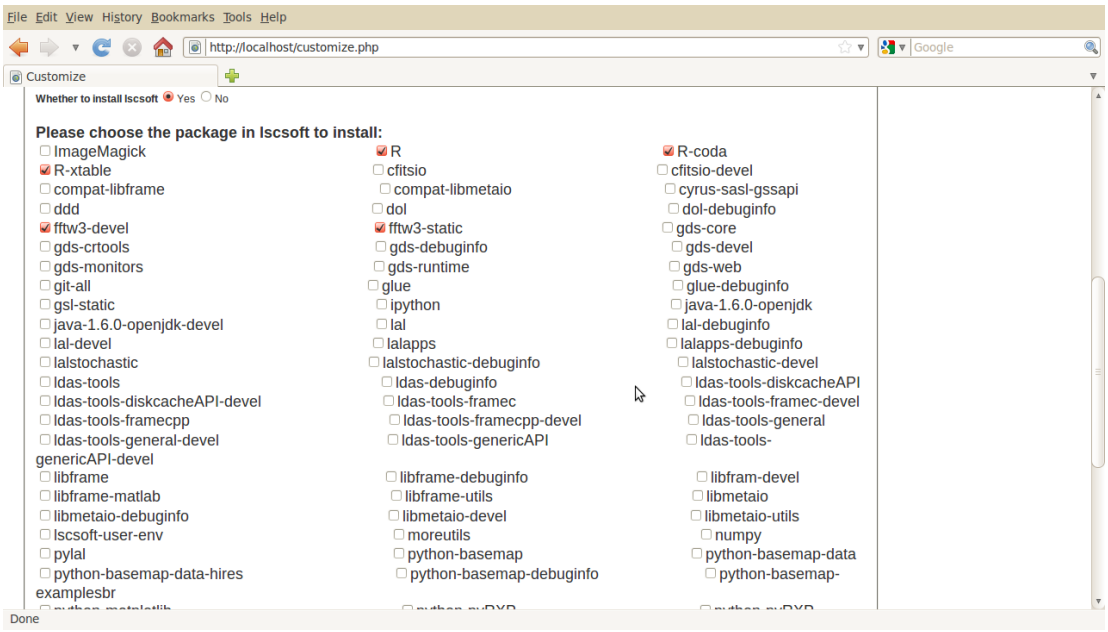

图3.16 定制页面（2）

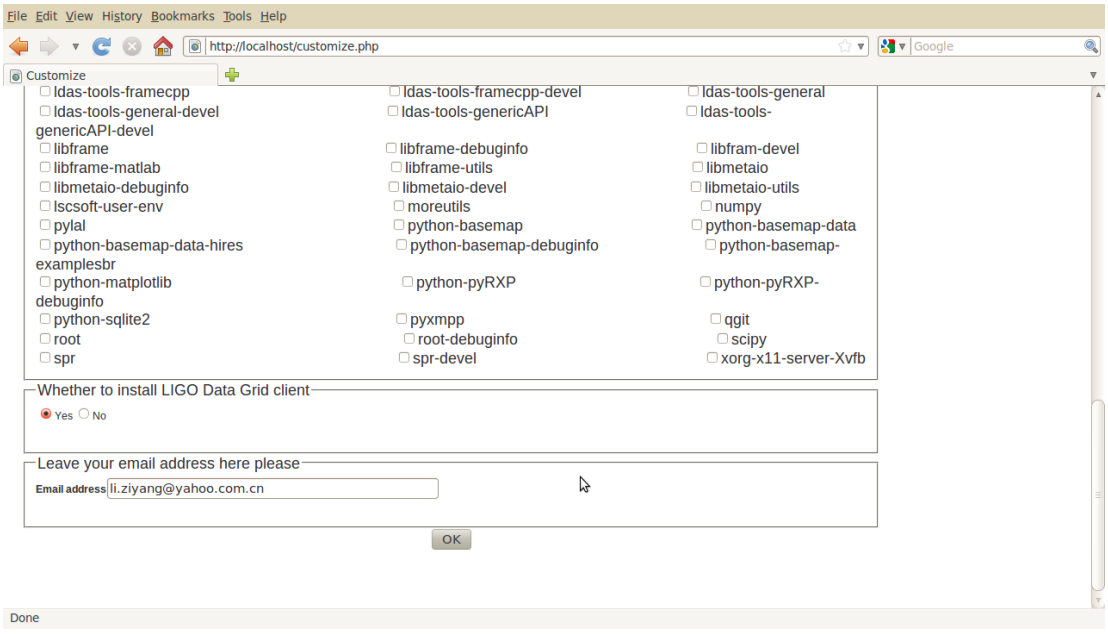

图3.17 定制页面（3）

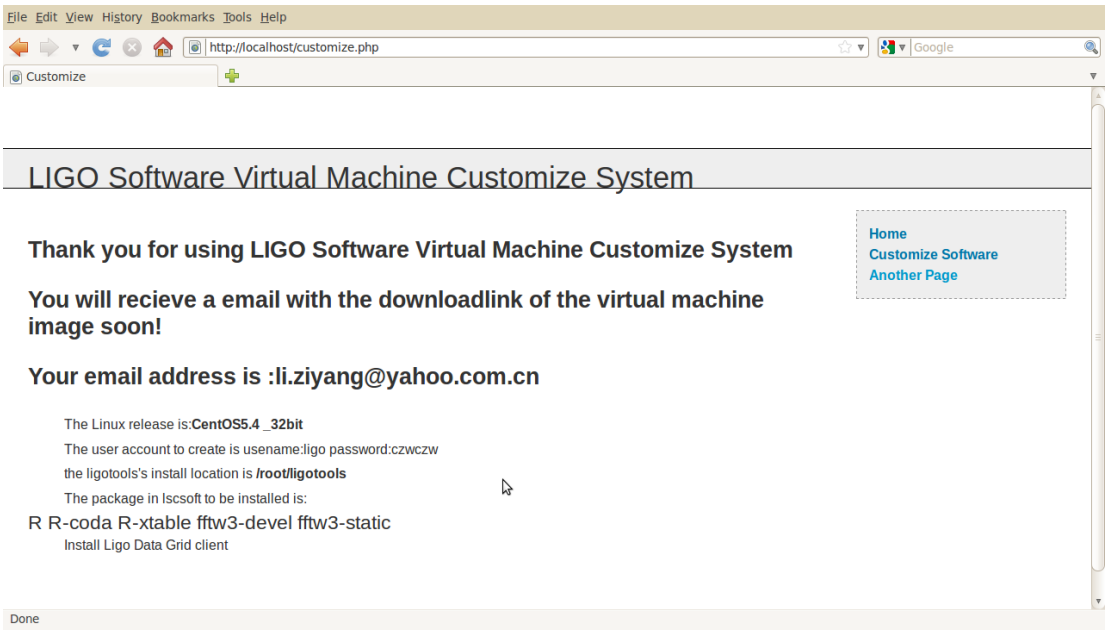

图3.18 软件定制信息显示页面

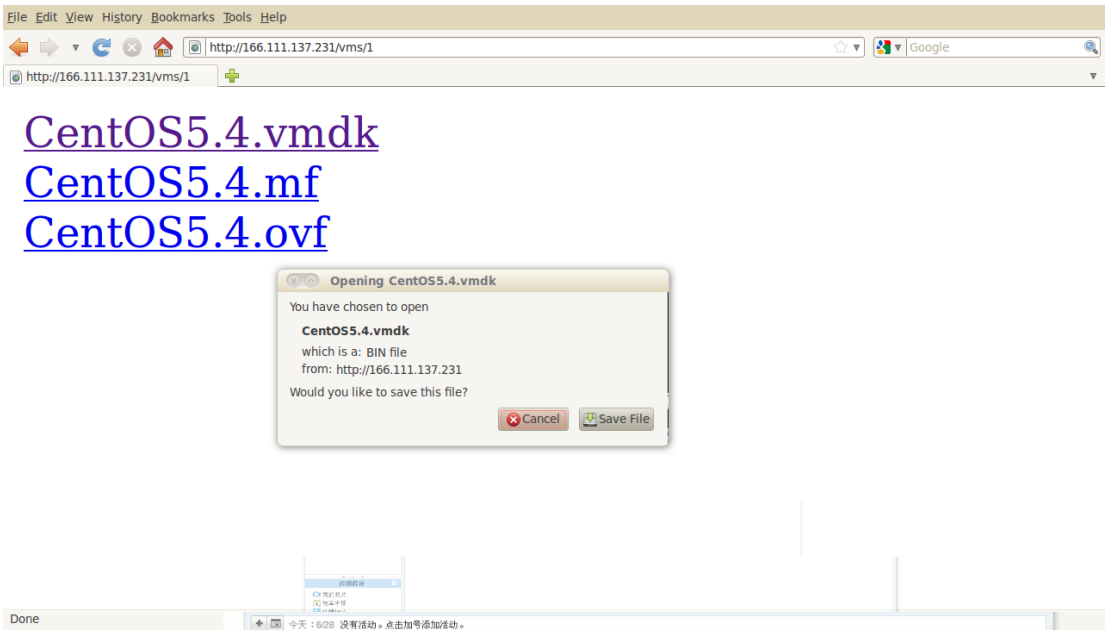

图3.19 用户下载页面

3.5 小结

本章详细介绍了基于虚拟机的 LIGO 软件定制服务系统的具体实现，概要介绍了 Expect 脚本编程方法，详细描述了系统准备过程，系统文件结构，以及 Web 页面和系统各脚本的流程。最终给出系统实现的 Web 页面效果图。

第4章 总结

4.1 课题成果总结

该系统的实现涉及到 Web 服务器搭建，页面编程，Shell 脚本编程，Expect 交互式工具，虚拟机软件的操作，Linux 系统设置等多种脚本编程语言和工具的使用。总结起来，各种工具的特点如下：

（1） Apache 是一种简单而有效的 Web 服务器软件，它由 Apache 组织进行开发和维护，是完全开源且免费的。使用 Apache 可以很方便地配置出一个服务器来。

（2） PHP 是一种广泛使用的通用脚本语言，特别适用于 Web 开发，并且可以嵌入在 HTML 中，进行动态页面编程。

（3） Expect 是非常实用的交互式开发工具，通过 Expect 可以实现与终端内程序进行通讯，对一些缺乏编程功能的工具，如 ftp，ssh，telnet 等进行扩充，使其具有自动运行的能力。

（4） Virtualbox 是由 Sun 公司提供的一款免费的虚拟机软件，封装良好，几乎所有的虚拟机操作都可由命令行完成。

（5） Linux 操作系统具有许多 Windows 操作系统所无法比拟的优势，其发行版众多，用户可根据自身喜好和需求自主选择。

（6） Linux Shell 是一种强大的脚本工具，通过 Shell 可以将 Linux 的命令行进行整合进而开发出强大的功能来。

本论文介绍了 LIGO 及 LIGO 软件，讲述了上述工具的使用方法，并使用这些工具搭建出 LIGO 虚拟机软件定制系统，详细描述了系统所有脚本的流程示意图。

利用上述工具，在本系统开发过程中，我主要完成以下几方面工作：

（1） 对项目需求进行整体分析，在解决方案中，针对各项需求选用相应的工具。采用模块化设计，系统分为三部分，一为 Web 服务器，二为系统运行脚本，三为系统维护脚本。

（2） 搭建 Web 服务器，使用 PHP 进行 Web 页面开发，满足功能需求。
（3） 应用 Expect 进行交互式脚本编程，实现自动登录，自动执行软件安装命令。
（4） 使用 Shell 脚本整合系统功能，将 Web 页面，Expect 脚本等模块有机整合起来，进而形成一个整体系统。
（5） 构建后台数据库，保存用户使用记录。
（6） 完成整个系统搭建及测试。

4.2 未来工作展望

未来该系统可以进行的改进主要有以下几方面：
（1） 在现有工作的基础上，在已有的 Expect 交互式脚本的基础上，增加对更多的操作系统镜像和更多软件的支持。
（2） 增加对用户使用情况的统计分析的功能，统计出用户偏好于定制哪些软件，对系统进行优化。
（3） 增加异常处理能力，对系统进行压力测试，提高系统鲁棒性。

插图索引

表格索引

参考文献

[1] http://www.ligo-la.caltech.edu/LLO/overviewsci.htm

[2] http://www.ligo.caltech.edu/

[3] http://www.ligo.org/

[4] http://en.wikipedia.org/wiki/LIGO

[5] http://www.ldas-sw.ligo.caltech.edu/ligotools/

[6] https://www.lsc-group.phys.uwm.edu/daswg/download/repositories.html

[7] http://en.wikipedia.org/wiki/Cluster_(computing)

[8] http://en.wikipedia.org/wiki/Virtual_machine

[9] http://www.ubuntu.com/

[10] http://expect.nist.gov/

[11] http://www.virtualbox.org/

[12] http://zh.wikipedia.org/zh/Expect

[13] 陈宗斌等译.《PHP6 与 MySQL5 基础教程》[M].人民邮电出版社.2008 年 11 月

[14] 陈宗斌等译.《PHP 与 MySQL 基础教程》（第二版）[M].2007 年 5 月

[15] 张颖等译.《PHP 实战》[M].人民邮电出版社.2010 年 1 月

[16] 宗杰，马国强，刘冉.《PHP 网络编程》[M].电子工艺出版社.2008 年 6 月

[17] 孟庆昌，牛欣源.《Linux 教程》（第二版）[M].电子工艺出版社.2007 年 3 月

[18] 李善平，施伟，林欣等译.《Linux 教程（LINUX: THE TEXT BOOK）》[M].2005 年 6 月

[19] https://www.lsc-group.phys.uwm.edu/lscdatagrid/doc/installclient.html

致谢

感谢我的指导老师曹军威老师，感谢他百忙之中给我的耐心指导，他严谨细致、一丝不苟的作风一直是我工作、学习中的榜样，老师平易近人、循循善诱的品德时时激励我在今后的生活学习中不懈努力。在此我向老师表达最诚挚的感谢！并祝老师身体健康，合家欢乐。

感谢指导我的李俊伟师兄，感谢他在我整个毕业设计中给予的帮助和指导，他知识渊博、态度严谨，更有平易近人、助人为乐的优良品德，在与之相处的时光里，师兄在学习上给了我很大的帮助及指导，从他身上我学到的不仅是文化知识，更多的是做人做事的态度和品德，这将是我未来的研究生生活的良好基础。

感谢实验室的王震，张帆，张文师兄，他们不仅在科研上以认真严谨的态度激励着我努力学习，而且他们在所里营造出的温馨和睦的气氛也深深的感染着我。

感谢辛勤养育我的父母。在我求学期间，他们始终给予我最大的支持和鼓励，使我战胜各种困难，顺利完成学业。希望我的进步能给他们带来喜悦和安慰。

最后衷心感谢在百忙之中评阅论文和参加答辩的各位老师！

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：李荣阳　　日期：2010.7.1

附录 A 外文资料的书面翻译

Intel-虚拟技术

Rich Uhlig，Gil Neiger，Dion Rodgers，Amy L. Santoni，Fernando C.M. Martins，
Andrew V. AndersonSteven M. Bennett，Alain Kägi，Felix H. Leung，Larry Smith
Intel Corporation

一旦限制到特定的服务器和大型机系统，虚拟化现在可以被基于Intel 架构硬件的现成的系统支持。Intel 虚拟技术为处理器虚拟化提供硬件支持，使得虚拟机监控软件（virtual machinemonitor software，VMMs）简单化。作为结果，虚拟机监控软件（VMMs）可以在保持高性能的同时广泛支持过去和未来的操作系统。

虚拟计算机系统的物理资源以达到改善分享和使用的目的这一想法已被很好的确定有十余年的时间了。完全的虚拟包括处理器、内存、I/O 设备在内所有的系统资源使得在一个物理平台上并行运行多种操作系统成为可能。

在一个非虚拟的系统里，一个单独的操作系统控制所有的硬件平台资源。一个虚拟的系统包括一个新的软件层——虚拟机监控层（VMM）。虚拟机监控层（VMM）的最主要职能是裁定的对主机平台底层物理资源的访问权限，以使得多种并行运行的操作系统（它们是虚拟机监控层（VMM）的客户）可以分享这些资源。虚拟机监控层（VMM）为每一个客户操作系统提供一系列虚拟的平台接口来组成一个虚拟机（VM）。

一旦限定到特定的、专有的、高端的服务器和大型机系统，虚拟化现在越来越被广泛地被利用，并且被基于 Intel 架构（Intel architecture，IA）硬件的现成的系统所支持。这种发展部分是由于基于 Intel 架构（IA）系统性能的稳定提升使得传统的虚拟化所造成的性能消耗得到缓解。其他因素包括新的富有创造性的用于解决 Intel 架构（IA）虚拟化所固有困难的软件技术的应用，以及在工业领域和学术界中所出现的虚拟化的新奇的应用。

虚拟化应用模式

虚拟化带来的传统的好处在于提高了大型机系统的利用率，易用性和可靠性。几个使用不同操作系统的用户可以很容易地分享一个虚拟的服务器，操作系统的升级可以分期的进行使故障停机的时间最短，并且客户软件错误可以隔离在错误发生的虚拟机内部。

这些好处在高端的服务器系统上被认为是非常有价值的，而最近的研究表明新出现的基于虚拟机监控（VMM）的产品预示着虚拟化的好处在大量的服务器和客户机系统中都有着广泛的应用。图 1 表明，虚拟化的三种功能能力可以包括大范围的用户。

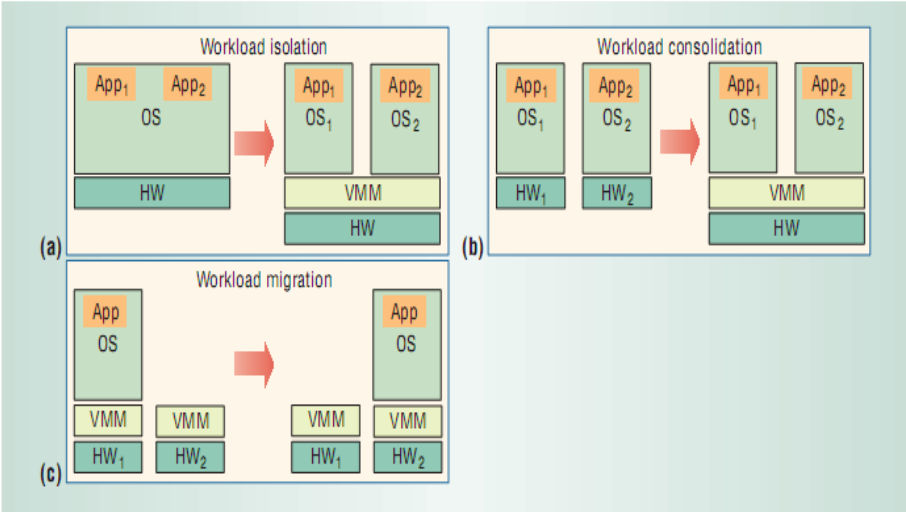

图 1.虚拟化功能.基于虚拟机监督系统的产品提供为大范围的虚拟化用户提供三种功能：（a）工作负载隔离，（b）工作负载整合，（c）工作负载迁移

工作负载隔离

通过将软件堆栈隔离在它们各自的虚拟机中，虚拟化能提高系统整体的安全性和可靠性。安全性可以被提高是因为非法入侵会被限制在它发生的那个虚拟机中，而可靠性可以被增强是因为一个虚拟机中的软件错误不会影响到其他的虚拟机。

Thomas Bressoud 和 Fred Schneider 通过在两个独立的虚拟机里运行完全相同的工作负载用来从系统错误中恢复出来，以测试虚拟技术所能达到的对系统错误容忍程度。Terra 和 ReVirt 项目是最近用虚拟技术来提高安全性的学术探索。系统软件的隔离特性在微软的 NGSCB（NextGeneration Secure Computing Base）和 VMware 的 ACE（Assured Computing Environment）中扮演中重要角色。

工作负载整合

由运行单一操作系统和单一应用负载的服务器比如 Web 主机或文件服务器，可造成不均衡为成分利用的数量增加，使得公司数据中心遭遇挑战。虚拟化可将个别的独立的工作量加给一个单独的物理平台，而减少总体消耗。

升级管理是技术主管们关心的另一个问题。当新的硬件或新的 OS 版本变得可用的时候，与老版本软件的不兼容问题经常会成为造成整个公司全部升级的开端。虚拟化可以通过允许同时运行老的和新的系统，来缓解这个矛盾。

在虚拟机中植入特定的系统功能可以方便客户使用。例如，将所有网络流量路由到一个网络管理虚拟机上，这个虚拟机可以提供一种“断路器”的能力，它可以把感染病毒的客户机从公司内部网中断开。

工作负载迁移

通过将客户的状态封装在一个虚拟机里，虚拟化可以将客户的工作与它正在运行的硬件解耦而移植到不同的平台上。

除了使硬件维护更容易外，虚拟机迁移可以由负载平衡或代理故障预测来自动触发。这种机制可以在较低经营成本的情况下，提高服务的质量。Xen 和 Internet Suspend-Resume Project 在服务器端和客户端都已经展示工作负载迁移，这种技术形成诸如 VMware 的 VMotion 之类的商业产品。

Intel 架构的纯软件虚拟化

已有的和新兴的应用对在服务器和客户端计算系统中虚拟化技术提供了强大的支持。不幸的是，基于 IA-32 和安腾架构的系统为这种支持强加了许多挑战。软件技术就是为了其中的一些挑战而存在的。

虚拟 Intel 架构的困难

Intel 微处理器提供基于两位的特权级别的保护，使用 0 表示最有特权的软件，而 3 表示最没有特权的。特权级别决定了如控制基本 CPU 功能等的特权指令能否被正确执行；它同时控制基于处理器页面表和 IA-32 架构段寄存器的地址空间的访问权限。如图 2 所示，大多数 Intel 架构软件（IA）使用特权等级只有 0 到 3。

一个操作系统要控制 CPU，它的一些元素必须以特权等级 0 运行。因为 VMM 不能让客户系统如此控制 CPU，客户系统不可以以特权等级 0 运行。这样基于 Intel 架构的虚拟机就必须使用一种叫“环路特权解除”（ring deprivileging）技术，这种技术可以以大于 0 的特权等级运行所有的客户软件。虚拟机通过在特权等级 1（0/1/3 模式）或在特权等级 3（0/3/3 模式）运行客户操作系统来解除它的等级 0 的特权。图 2b 和 2c 显示了这两种选择。机关 0/1/3 模式可以支持简单的虚拟机监控软件（VMMs），但是它不能在 Intel32 位架构（IA-32）上为客户提供 64 位模式。64 位模式是 Intel 的扩展 64 位内存技术（Extended Memory 64 Technology，EM64T）的一部分，将 IA-32 扩展为 64 位。

环路特权解除技术（ring deprivileging）引发了许多虚拟化困难。

环路混淆（ring aliasing）　环路混淆是由软件没有在它原来指定上的特权等级上运行所引来的。IA-32 上的一个例子是 PUSH 指令，PUSH 指令当与系统寄存器执行时将它的操作数压入堆栈（在当前特权权限下）。同样地，安腾架构的 br.call

将当前的特权等级保存在可以在任何特权等级被读取的 PFS 寄存器的 ppl 位。在其他情况下，客户操作系统可以很容易地判定它没有在特权等级 0 上运行。

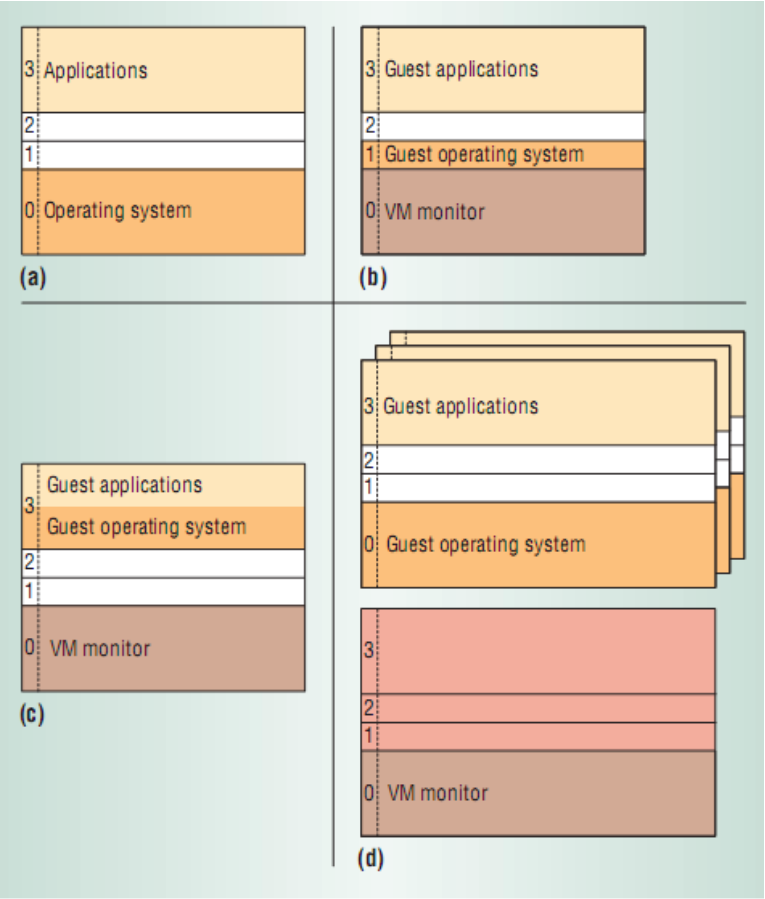

图 2.虚拟化方法.（a）系统操作位于等级 0，应用软件位于等级 3 的非虚拟操作系统；（b）“环路特权解除”（ring deprivileging）技术的 0/1/3 模式；（c）“环路特权解除”（ring deprivileging）技术的 0/3/3 模式；（d）使用 VT-x 或 VT-i 代替强迫解除客户软件特权的系统。

地址空间压缩（Address-space compression）以线性地址而著称的 IA-32，操作系统希望拥有处理器全部虚拟地址空间的访问权限。VMM 必须在客户虚拟地址空间中为它自己保留一部分。VMM 可以在客户地址空间里完全运行，它可以很容易地访问客户端的数据，尽管 VMM 的指令和数据可能用掉了大量的客户地址空间。

作为一种选择，VMM 可以在单独的地址空间中运行，但是在这种情况下的 VMM 的控制结构必须用最少的客户虚拟地址空间来管理客户软件和 VMM 之间的转换。（IA-32 中这些结构包括存在于线性地址空间中的 IDT 和 GDT。安腾架构中这些结构包括存在于虚拟地址空间的 IVT）

VMM 必须阻止客户访问用户地址空间中 VMM 所占用的部分。另外，如果客户系统可以写这一部分地址空间，则 VMM 的公平性就会受到削弱；或者客户系统可以探测出它是运行在虚拟机中如果它可以读这一部分的话。客户系统试图访问那些由 VMM 占据的地址空间，VMM 必须模拟或者以其他方式支持他们。术语*地址空间压缩(address-space compression)*指的是保护这部分虚拟地址空间和支持客户系统他们所带来的问题。

特权状态的无错访问（Nonfaulting access to privileged state） 基于特权等级的保护机制阻止非特权软件访问 CPU 状态的某一部分。大多数情况里，尝试直接访问将得到错误的结果，允许 VMM 仿真客户指令的需求。然而，IA-32 和安腾架构都包括存取特权状态的指令，并且当没有足够权限访问时不会产生错误。例如，IA-32 寄存器 GDTR，IDTR，LDTR 和 TR 包含控制 CPU 操作的数据结构的指针。软件只有在特权等级 0 下才可以执行写或装在一些寄存器（LGDT，LIDT，LLDT 和 LTR）。但是，软件在任意特权等级下可以执行读或存储另一些寄存器（SGDT，SIDT，SLDT 和 STR）。如果 VMM 用意想不到的值维护这些寄存器，客户系统通过使用最近的指令可以判断它没有完全掌控 CPU。

另一个例子适用于安腾页面表地址寄存器（PTA），该寄存器包含有虚拟哈希页面表（VHPT）的基地址。mov 指令时访问 PTA 寄存器的正常方式，软件可以以特权等级 0 执行。然而 thash 指令间接地全部或部分暴露了 VHPT 的基地址，软件可以以任何特权等级来执行。如果 VMM 以一个与客户操作系统期望的不同地址来维护 VHPT，客户操作系统使用 thash 指令可以判断出它没有完全控制 CPU。

客户转换的负面影响（Adverse impacts on guest transitions） 环路特权解除技术干涉 IA-32 和安腾架构的设备功能可以加速传送和处理操作系统软件之间的转换。IA-32 的 SYSENTER 和 SYSEXIT 指令支持低延迟的系统调用。SYSENTER 影响到特权等级 0 的转换，而 SYSEXIT 在执行超过特权等级的指令时会失败。这样环路特权解除技术有以下启示：

- 客户应用完成 SYSENTER 后将导致场景转换到 VMM 而非客户操作系统。这样，VMM 必须仿真客户所执行每一个 SYSENTER。
- 客户操作系统执行 SYSEXIT 将导致给 VMM 的一个错误。这样，VMM 必须仿真客户所执行每一个 SYSEXIT。

安腾架构通过提供中断和中断环境的信息提供有效率的中断处理操作。这些数据不是记录在内存中，而是一系列的中断控制寄存器中的。，处理器为了保护系统的完整性，将以产生错误回应以非特权等级 0 访问这些寄存器的操作。典型地，每一个中断处理者读取这些中断处理寄存器。如果这种访问给 VMM 生成了一个错误，那么这些中断处理的效率将受到严重拖累。

中断虚拟（Interrupt virtualization）提供外部中断尤其是关于中断屏蔽的支持，给 VMM 的设计展现出一些比较特别的麻烦。IA-32 和安腾架构都提供了遮蔽外部中断的机制，防止在操作系统还没准备好的时候将中断传递给它。IA-32 使用 EFLAGS 寄存器中的中断标志位（interrupt flag，IF）来控制中断屏蔽；安腾架构使用 PSR 中的 i 位来提供这种功能。

VMM 很可能会管理外部中断而拒绝客户软件控制中断屏蔽的能力。已经存在的保护机制允许这样的拒绝客户控制。在环路特权解除的环境下，客户试图控制中断屏蔽会失败。这种失败会引发问题，因为一些操作系统经常会屏蔽中断和解除屏蔽，截取客户每一个这样的请求将会明显地影响系统的性能。

Intel 架构术语表

IA-32 和安腾架构分别有自己独特的指令，寄存器和表格，其中一些如下所列。

IA-32 术语

CPUID：CPU 识别指令

CR：控制寄存器：CR0，CR3（页面表基地址，控制），CR4 和 CR8（当前任务优先级）

CS：当前程序段的段寄存器；在某些模式下，它的低两位是它的当前特权等级。

DR：调试寄存器

EFLAGS：标志位寄存器的 32 位版本，既包含算术标志位又包含屏蔽中断的中断标志位（IF）

GDT：global descriptor table，全局描述表，包含有可以被装载到段寄存器 LDTR 和 TR 的描述元

GDTR，IDTR，LDTR，TR：引用 GDT，IDT，LDT 和 TSS 的寄存器

HLT：断点指令，停机指令

IDT：中断描述表，控制异常和中断向软件处理者的传递

IF：EFLAGS 寄存器中控制中断屏蔽的标志位

INVLPG：使 TLB 入口指令无效

LDT：本地描述表，包含有可以装载到段寄存器的描述符

LGDT，LIDT，LLDT，LTR：向 GDTR，IDTR 和 TR 写的指令

MOV：移动指令，允许读写控制寄存器和调试寄存器

MWAIT：监控等待指令

PUSH：将操作数压入堆栈

RDMSR，WRMSR：读写模块寄存器的指令

RDPMC：读性能监视计数器的指令

RDTSC：读时间戳计数器的指令

segment register：段寄存器，控制将逻辑地址翻译到物理地址的寄存器

SGDT，SIDT，SLDT，STR：从 GDTR，IDTR 和 TR 读取的指令

SYSENTER，SYSEXIT：快速系统指令中的快速系统调用和快速返回指令

TSS：任务状态段，在其他事情中，当前 TSS 控制软件访问 I/O 端口的能力。

安腾术语

br.call：转移指令通常用来产生一个有条件的过程调用

i：PSR 中控制中断屏蔽的位

IVT：中断矢量表，控制异常和中断向软件处理者的传递

mov：移动指令，允许读写控制寄存器（包括 PTA）

PFS：以前动作状态寄存器

ppl：PFS 中的特权等级位区域

PAL：页面表地址寄存器

rfi：从中断指令中返回

thash：翻译散列入口地址指令

VHPT：虚拟散列页面表，控制从虚拟地址到物理地址的翻译

尽管不通过拦截客户每一次请求来阻止其修改中断屏蔽是可能的，当 VMM 有“虚拟中断”传递给客户时依然是有困难的。一个虚拟中断只有在客户没有屏蔽中断时才可以传递给客户系统。为了适时的传递虚拟中断，VMM 应该拦截一些但不是全部的客户屏蔽中断的请求。这样做会使 VMM 的设计变得显著复杂起来。

环路压缩（Ring compression）环路特权解除技术使用基于特权等级机制来保护 VMM。IA-32 有两种这样的机制：段限制和分页。段限制不适用于 64 位模式，所以分页就必须以这种方式来使用。由于 IA-32 分页不能区分 0 到 2 的特权等级，客户操作系统必须在特权等级 3 运行。这样，客户操作系统将会和客户应用软件在同一特权等级上运行，而不能获得保护。这个问题被称为环路压缩。

访问隐藏状态（Access to hidden state）CPU 的一些状态没有在任何软件可以访问的寄存器中展现出来。比如隐藏的段寄存器描述缓存。段寄存器在这个缓存里装载引用描述符（来自 GDT 或 LDT），它将不会被修改如果软件稍后会写入到描述表中的话。当切换虚拟机时，IA-32 不提供保存并恢复或保留客户环境的隐藏元素的机制。

软件寻址虚拟化的困难

为了应对 IA-32 或安腾架构所带来的虚拟化困难，VMM 设计者们研究了富有创造性的修改客户软件的解决方案。Denali 和 Xen 通过使用一种叫“半虚拟化”的技术实现源文件级别的修改。这些 VMM 的作者修改客户操作系统内核和设备驱动，创造出更容易虚拟的借口。

半虚拟化提供了高性能并且不需要使用户的应用做出改变。一个不利之处在于，半虚拟化支持的操作系统范围有限。例如，Xen 一般不能支持像 Microsoft Windows 这样的它的开发者没有做出相应修改的操作系统。

VMM 通过直接修改客户操作系统的二进制文件可以支持老的操作系统。包括 VMware 和微软的 Virtual PC，Virtual Server 在内的虚拟机监控软件使用这种二进制翻译技术。这些虚拟机软件尽管比半虚拟化的虚拟机有着更高的性能消耗，但是它们支持很广范围内的操作系统。

Intel 虚拟化的核心目标是消除对 CPU 半虚拟化和二进制翻译技术，从而使 VMMs 在保持高性能的同时，可以支持广范围内的未做更改的客户操作系统。

Intel 虚拟技术

Intel 虚拟技术包括支持 IA-32 虚拟处理器的 VT-x 和支持安腾架构的 VT-i。

VT-x 架构概览

VT-x 在 IA-32 上扩展了两种新形式的 CPU 操作：VMX 根运作（VMX root operation）和 VMX 非根运作（VMXnon-root operation）。VMM 以 VMX 根运作运行，而其客户软件以 VMX 非根运作运行。这两种形式的操作都支持所有四个特权等级，允许客户操作系统在原定的特权等级上运行，为 VMM 提供灵活的多种特权等级使用方式。VMX 根运作类似于没有 VT-x 的 IA-32。以 VMX 非根运作运行的软件是以特定方式的无视特权等级的特权解除。

VT-x 定义了两种新的转换：从 VMX 根运作到 VMX 非根运作，也就是从 VMM 到客户的转换，叫做虚拟机进入（VM entry），相反的转换叫做虚拟机退出（VM exit）。

虚拟机控制结构（the virtual-machine control struture，VMCS）是一种新的管理虚拟机进入，VM exit 和处理器在 VMX 非根操作下行为的数据结构。VMCS 逻辑上分为几部分，其中两部分是客户状态区域和主机状态区域。这些区域包含处理器状态的不同部分相对应的区域。VM entry 时从客户状态区域装载处理器状态。VM exit 时保存处理器状态到客户状态区域，并且从主机状态区域装载处理器状态。

处理器行为在 VMX 非根运行下发生实质变化。最重要的是，很多指令和事件引发虚拟机退出。一些指令不能在 VMX 非根运作下执行是因为它们会导致无条件的 VM exit。这些指令包括 CR3，RDMSR 和 WRMSR 的 CPUID 和 MOV 指令。其他指令，中断和异常使用 VMCS 中的虚拟机执行控制区域可以配制成引起有条件的 VM exit。

虚拟机执行控制区域（VM-execution control fields）虚拟机执行控制区域允许 VMM 灵活地指定引发 VM exit 的指令和事件。以下指令有单独的控制：HLT，INVLPG，MOV，CR8，MOV DR，MWAIT，RDPMC 和 RDTSC。这些控制支持各种各样的虚拟化控制策略。附加的控制允许有选择的保护 CR0，CR3 和 CR4 。

VT-x 有两种支持中断虚拟化的控制。外部中断退出控制位置位的情况下，所有外部中断引发 VM exit；作为附加的，客户不能屏蔽中断。在中断窗口退出控制位置位的情况下，当客户软件认为它准备好接受中断时，VM exit 发生。

为了支持 VMM 的灵活性，VT-x 包括位映像允许 VMM 有选择地关注一些 VM exit 的起因。它们中的一个是包含有 IA-32 的 32 个条目的异常位映像。它允许 VMM 指定哪个异常应当引发 VM exit，而哪个不行。另一个位映像允许 I/O 指令的每端口控制。

VMCS 细节。客户状态区域包含有与 VMCS 有关的虚拟 CPU 的状态。它包括与 IA-32 控制处理器操作寄存器（如段寄存器，CR3 和 IDTR）相一致的区域。

作为附加地，客户状态区域包括无记录的 CPU 状态相一致的区域，如段寄存器的描述缓存。这些内容使得 VMM 可以在一个 VM 没有运行时记录它们的值，并且在 VM 重启时恢复它们。

VMM 通过物理地址而不是线性地址引用 VMCS。这样就消除了定位 VMCS 在客户线性地址空间（与 VMM 线性地址空间不同）位置的麻烦。

VM entry 和 exit。虚拟机进入从 VMCS 的客户状态区域装载处理器状态。为了遵循这个装载，VMM 可以通过注入一个中断或异常来随意地配置 VM entry。CPU 使用客户 IDT 响应这种注入，就好像注入的事件是在 VM entry 之后立即发生的一样。这种特性去掉了 VMM 仿真传递这些事件的需要。

VM exit 保存处理器状态到客户状态区域，从主机状态区域装载处理器状态。所有 VM exit 使用一个通用的到 VMM 的入口点。为了简化 VMM 的设计，每一个 VM exit 保存到 VMCS 细节信息会具体说明退出的原因；许多退出也记录退出条件来提供更深层的资料。

例如，如果 MOV CR 指令引发一个 VM exit，则退出原因将显示“访问控制寄存器”；退出资格将显示（1）控制寄存器的名称（例如，CR0）；（2）MOV 是到还是从控制寄存器来；（3）哪一个一般寄存器是该指令的来源或目的地。

VM entry 和 VM exit 都装载（页面表层级的基地址）。这意味着 VMM 和客户在不同的线性地址空间中运行。

VT-i 架构概览

VT-i 由安腾处理器硬件扩展和处理器抽象层（PAL）固件组成。

处理器状态位 PSR.vm。重要的硬件扩展是在处理器状态寄存器（PSR）中新增了一位（vm）。VMM 本身运行时 PSR.vm=0；它运行客户软件时 PSR.vm=1。所有四个特权等级可以无视 PSR.vm 的值运行；客户软件可以在原来的特权等级上运行，而 VMM 有使用多种特权等级的自由。当 PSR.vm=0 时，处理器操作类似于没有 VT-i 情况下的操作，一些没有特权的指令，例如 thash 引起新的虚拟化错误。

当所有中断通过 IVT 传递时，PSR.vm 被清零；这样 VMM 或 PAL 处理所有的，甚至是那些属于客户软件的中断。VMM 或 PAL 通过使用 rfi 指令可以将 PSR.vm 置为 1 来返回客户软件。VT-i 增加一个新的指令叫 vmsw（virtual machine switch），它可以以最小的费用来定义 PSR.vm 位，减少在共有的虚拟环境中客户软件与 VMM 之间切换的延时。

PSR.vm 也控制着对软件有效地虚拟地址位的个数。当 VMM 运行的时候，也就是 PSR.vm=0 的时候，所有执行虚拟地址位是可见的。当客户软件运行时，也就是 PSR.vm=1 时，最重要的虚拟地址位是不可见的，如果该位被使用了则会抛出一个异常。

IVT 向量。为了促进 VMM 有效率地处理转换，VT-i 为 IVT 增加了两个新的向量。VMM 使用虚拟化向量来配置处理器以使用处理器分组寄存器（processor-banked registers）中的两个来识别虚拟化错误原因和错误操作码。通过虚拟外部中断向量，VMM 可以使用 PAL 服务来记录未处理的虚拟中断。如果 VMM 已经注册了一个中断且解除屏蔽的客户执行了相应操作，则控制转移给虚拟外部中断向量。

PAL 固件层扩展。VT-i 包含对 PAL 固件层的扩展为 VMM 提供一致的编程接口，尽管硬件通过处理器的更新换代而没有同样的同一地执行。PAL 扩展包括一系列新的程序，高频率 VMM 操作的 PAL 服务的扩展和虚拟处理器描述（VPD）表。

VT-i 定义 PAL 程序为建立和销毁虚拟机环境，初始化和终结虚拟处理器状态，保存和恢复虚拟处理器状态。这些程序如其他的 PAL 程序遵守相同的调用协定。

VT-i 将 PAL 为虚拟化提供的接口称为服务。为了减少性能消耗，PAL 服务针对 VMM 专门使用了新的调用方式。PAL 服务提供了包括同步客户影寄存器和 VPD，保存和恢复虚拟处理器状态的子集，从 VMM 转换后恢复执行客户软件在内的几个功能。

PAL 固件和 VMM 都可以访问位于内存中的 VPD。VPD 包含有虚拟处理器的配置参数和虚拟处理器状态中影响运行状态的部分。例如，虚拟处理器的控制寄存器的值位于 VPD 中。VPD 包含两个配置趋于，允许 VMM 定义虚拟化环境。

• 虚拟化加速区域（virtualizationacceleration field）为特定的资源和指令加速虚拟化。例如，它允许优化使得当客户用错误的处理程序访问中断控制寄存器时减少 VMM 需要处理的转场的次数。

•虚拟化禁止区域（virtualizationdisable field）禁止特定资源或指令的虚拟化。例如，VMM 可以使用户直接访问专门用于外部中断的控制寄存器。

用 VT-x 和 VY-i 解决虚拟化困难

VT-x 和 VT-i 通过允许客户软件在它原来的特权等级上运行来解决虚拟化的问题。客户软件不是通过特权等级被限制的，而是由于它以 VMV 非根运作（VT-x）下运行的或者在 PSR.vm=1 下运行（VT-i）的。图 2d 描述了这种用法。

地址空间压缩（Address-space compression）

VT-x 和 VT-i 为解决地址空间压缩问题提供两种不同的技术方案。VT-x 中，每一次 VMM 和客户软件之间的转场会改变线性地址空间允许客户软件完全使用它的地址空间。VMX 转场是由 VMCS 控制的，VMCS 是存在于物理空间而不是线性地址空间的。

VT-i 中，VMM 有一个客户软件无法使用的虚拟地址位。VMM 可以通过拦截客户对 PAL 程序的调用隐藏硬件对该位的支持，来报告可以使用的虚拟地址位。作为结果，客户不会期望使用这个最高位，允许 VMM 专有一半的虚拟地址。

环路别名（ring aliasing）和环路压缩（ring compression）

VT-x 和 VT-i 消除了环路别名的问题，因为它们允许 VMM 以客户软件原来的特权等级上运行它们。PUSH（CS 的）和 br.call 这样的指令不会暴露出软件是在虚拟机上运行的。VT-x 也消除了当客户操作系统与客户应用在同一特权等级上运行时出现的环路压缩问题。

特权等级的无错访问（Nonfaulting access to privileged state）

VT-x 和 VT-i 以两种方式避免特权等级的无错访问问题：通过增加对这种进入VMM转场的诱因的支持的方式和通过增加支持使得这种进入状态对VMM变得不重要的方式。

基于 VT-x 的 VMM 不需要控制客户特权等级，VMCS 控制中断和异常的处理。这样，它允许它的客户访问 GDT，IDT，LDT 和 TSS。VT-x 允许客户软件在特权等级 0 运行 LGDT，LIDT，LLDT，LTD，SGDT，SIDT，SLDT 和 STR 等指令。

VT-i 中，thash 指令引发虚拟化错误，为 VMM 提供一个隐藏对 VHPT 基地址进行任何修改的机会。

客户转变（guest transitions）

客户软件在非特权等级 0 下不能使用 IA-32 指令 SYSENTER 和 SYSEXIT。在 VT-x 下，客户操作系统可以在特权等级 0 下运行，允许使用这些指令。

在 VT-i 下，VMM 可以使用 VPD 中的虚拟加速区域来表明客户软件可以不用向 VMM 发出请求的情况下读或写中断控制寄存器。VMM 可以在任何虚拟中断被传送前设置这些寄存器的值，并且可以在客户中断处理程序返回前修改它们。

中断虚拟化（Interrupt virtualization）

VT-x 和 VT-i 都为中断屏蔽虚拟化提供了明确的的支持。

VT-x 包含一个外部中断退出虚拟机执行控制机制。当这个控制位被置为 1 时，VMM 在不用附加的控制客户修改 EFLAGS.IF 位的请求的情况下阻止客户控制中断屏蔽。VT-i 中的虚拟机加速区域可以阻止客户影响中断屏蔽，并且避免了每一个访问 PSR.i 位请求引发的向 VMM 的转换。

VT-x 也有中断窗口退出 VM 执行控制的机制。当这种控制位被置为 1 时，只要客户软件转备好接受中断就会发生 VM exit。VMM 可以设置这个控制位当它由一个虚拟中断要传递给客户的时候。VT-i 有使得 VMM 可以注册未处理的虚拟中断的 PAL 服务。当客户软件准备好接受这个中断时，这个服务通过新的虚拟外部中断向量将控制传递给 VMM。

访问隐藏状态（access to hidden state）

VT-x 内 VMCS 区域中的客户状态区域与 CPU 状态相一致，不会在任何软件可达的寄存器中展现出来。处理器每次 VM entry 从这些 VMCS 区域装载值，而每次 VM exit 时再保存回这些区域中。这为当 VMM 在运行或切换 VM 时保存状态提供了必要的支持。

当被限制在专有的服务器和大型计算系统时，虚拟技术在服务器和客户端系统已有的和新兴的应用已经成为主流。

不管新兴的已存在的虚拟机用户条款的允诺，现在基于 IA-32 系统，很多问题阻碍虚拟化效率的提高。像二进制翻译和半虚拟化这样创造性的软件技术已经解决了部分问题，但是这些问题的范围意味着这些解决方案要么太过复杂缺乏鲁棒性，要么他们的能力不够完备无法运行未修订的老的操作系统。

VT-x 和 VT-i 是 Inel 虚拟技术第一个组成部分，一系列的处理器和芯片改革在基于 Intel 架构的客户端和服务器平台上成为可能。VT-x 和 VT-i 为 IA-32 和安

腾处理器虚拟化的固有问题提供了解决方案，使得在保持高性能的条件下更简单的，更健壮的，更安全的支持更广范围操作系统的 VMM 软件的出现成为可能。

[1] Intel Corporation. Intel Virtualization Technology. [J]. Computer. Published by the IEEE Computer Society. 2005

综合论文训练记录表

学生姓名	李紫阳	学号	2006011547	班级	自 61
论文题目	LIGO 软件虚拟机定制服务				
主要内容以及进度安排	第 1 周-第 3 周(3 月 1 日-3 月 21 日)：调研，准备开题。 第 4 周-第 5 周(3 月 22 日-4 月 4 日)：搭建 HTTP 服务器，实现最简单的功能。用户通过浏览器访问服务器，服务器执行相应脚本。 第 6 周-第 8 周(4 月 5 日-4 月 25 日)：完善丰富脚本，实现系统整体功能。 第 9 周-第 10 周(4 月 26 日-5 月 9 日)：系统测试，调试 BUG。 第 11 周-第 14 周(5 月 10 日-6 月 6 日)：撰写毕业论文 第 15 周-答辩(6 月 6 日-答辩)：准备答辩。 指导教师签字： 考核组组长签字： 2010 年 3 月 24 日				
中期考核意见	李紫阳同学工作进展顺利，基本完成了软件安装脚本等前期工作，希望进步完善系统搭建和开发。 考核组组长签字： 10 年 4 月 28 日				

指导教师评语	李紫阳同学的工作是为生物科学家提供一种简易的软件定制解决方案，运用Web和虚拟机技术，李紫阳的系统可以根据用户的定制，在线动态生成所需镜像，工作具有一定的创新性。 指导教师签字： 2010 年 6 月 29 日
评阅教师评语	李紫阳同学的本科毕业设计工作在题目选取上富有新意，设计与开发工作合理完整，毕业论文清晰完整，符合本科毕业标准。 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。 同意通过答辩。 答辩小组组长签字： 2010 年 6 月 29 日

总成绩：93

教学负责人签字：

2010 年 7 月 1 日

开题89(15%)，中期90(25%)、95(60%)