



Red Hat Enterprise Linux 5

Virtualization

A reference guide for virsch, xm, vmm and xend.

版 2.0

Red Hat Enterprise Linux 5 Virtualization

A reference guide for virsch, xm, vmm and xend.

版 2.0

Landmann

rlandmann@redhat.com

法律通告

Copyright © 2007 Red Hat Inc..

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

本指南介绍了如何使用 `virsh`、`xm`、`vmm` 和 `xend` 来配置、创建和监控红帽企业 Linux 5 上的虚拟客户机操作系统。如果你在《红帽企业版 Linux 虚拟化指南》发现了错误，或者想完善本指南，我们乐意接受你的建议！你可以在 Bugzilla (<http://bugzilla.redhat.com/bugzilla/>) 里提交关于红帽企业版 Linux 和虚拟化指南的报告。

目录

第 1 章 RED HAT 虚拟化系统体系结构	4
第 2 章 操作系统支持	5
第 3 章 硬件支持	6
第 4 章 RED HAT 虚拟化系统的系统要求	7
第 5 章 引导系统	8
第 6 章 配置 GRUB	9
第 7 章 引导客户机域	11
第 8 章 在引导时启动/停止域	12
第 9 章 配置文件	13
第 10 章 管理 CPU	14
第 11 章 移植域	15
第 12 章 配置网络应用	16
第 13 章 设置 DOMAIN0 的安全性	17
第 14 章 存储	18
第 15 章 用 VIRSH 管理虚拟机	19
15.1. 连接至监控程序	19
15.2. 创建虚拟机	19
15.3. 配置 XML 转储	19
15.4. 挂起虚拟机	19
15.5. 恢复虚拟机的运行	19
15.6. 保存虚拟机	20
15.7. 恢复虚拟机	20
15.8. 关闭虚拟机	20
15.9. 重新启动虚拟机	20
15.10. 终止域	20
15.11. 把域名转换为 DOMAIN ID	20
15.12. 把 DOMAIN ID 转换为域名	21
15.13. 把域名转换为 UUID	21
15.14. 显示虚拟机信息	21
15.15. 显示节点信息	21
15.16. 显示虚拟机信息	21
15.17. 显示虚拟 CPU 信息	22
15.18. 配置虚拟 CPU 关联	22
15.19. 配置虚拟 CPU 的数量	22
15.20. 配置内存分配	22
15.21. 配置最大内存数量	23
15.22. MANAGING VIRTUAL NETWORKS	23
第 16 章 用 XEND 管理虚拟机	25
第 17 章 用虚拟机管理者 (VIRTUAL MACHINE MANAGER) 管理虚拟机	27
17.1. 虚拟机管理者架构	27

17.2. OPEN CONNECTION 窗口	27
17.3. 虚拟机管理者窗口	27
17.4. 虚拟机 DETAILS 窗口	28
17.5. 虚拟机图形化控制台	28
17.6. STARTING THE VIRTUAL MACHINE MANAGER	29
17.7. 创建新的虚拟机	30
17.8. 恢复保存的虚拟机	39
17.9. DISPLAYING VIRTUAL MACHINE DETAILS	41
17.10. CONFIGURING STATUS MONITORING	44
17.11. 显示 DOMAIN ID	45
17.12. 显示虚拟机状态	46
17.13. 显示虚拟 CPU	47
17.14. 显示 CPU 的使用情况	48
17.15. 显示内存使用情况	49
17.16. MANAGING A VIRTUAL NETWORK	50
17.17. CREATING A VIRTUAL NETWORK	51
第 18 章 红帽虚拟化系统的故障解除	60
18.1. 日志文件概述和位置	60
18.2. 日志文件的描述	60
18.3. 重要的目录位置	60
18.4. 故障解除工具	61
18.5. 利用日志进行故障解除	62
18.6. 用串行控制台进行故障解除	63
18.7. 对半虚拟化客户机控制台的访问	63
18.8. 对完全虚拟化客户机控制台的访问	63
18.9. 实施 LUN 持久化	64
18.10. SELINUX 的相关事宜	65
18.11. 访问客户机磁盘映像里的数据	65
18.12. 常见的故障解除	66
18.13. 回路设备错误	67
18.14. 客户机创建错误	67
18.15. 串行控制台错误	67
18.16. 网桥错误	67
18.17. 笔记本配置	69
18.18. STARTING DOMAINS AUTOMATICALLY DURING SYSTEM BOOT	70
18.19. 修改 DOMAIN0	71
18.20. 客户机配置文件	71
18.21. 克隆客户机配置文件	72
18.22. 创建生成 MAC 地址的脚本	73
18.23. 配置虚拟机的 LIVE 移植	73
18.24. 翻译错误信息	73
18.25. 关于故障解除的在线资源	76
第 19 章 其他的资源	78
19.1. 有用的网站	78
19.2. 安装的文档	78
附录 A. 实验 1	79
附录 B. 实验 2	84
附录 C. REVISION HISTORY	86

第 1 章 RED HAT 虚拟化系统体系结构

红帽虚拟化系统是多层的，它由专用的红帽虚拟化组件驱动。红帽虚拟化系统可以支持多个客户机操作系统。每个客户机操作系统都运行在自己的域里。红帽虚拟化系统在虚拟机内调度虚拟 CPU 来最好地利用可用的物理 CPU。每个客户机操作系统都处理自己的应用程序。这些客户机操作系统也相应地调度每个应用程序。

你可以以两种方式部署红帽虚拟化系统：**完全虚拟化 (full virtualization)** 或**半虚拟化 (paravirtualization)**。完全虚拟化提供底层物理系统的全部抽象化，且创建一个新的虚拟系统，客户机操作系统可以在里面运行。不需要对客户机操作系统或者应用程序进行修改（客户机操作系统或者应用程序象往常一样运行，意识不到虚拟环境的存在）。半虚拟化需要对运行在虚拟机上的客户机操作系统进行修改（这些客户机操作系统会意识到它们运行在虚拟环境里）并提供相近的性能。完全虚拟化和半虚拟化都可以部署在你的虚拟化构架里。

第一个域，称为 **domain0 (dom0)**，在你引导系统时自动创建。Domain0 是专用的客户机，它拥有创建新域并管理它们的虚拟设备的管理能力。Domain0 处理物理硬件，如网卡和硬盘控制器。Domain0 也处理管理性的任务，如暂停、恢复执行或移植客户机域到其他虚拟机里。

The **hypervisor (Red Hat's Virtual Machine Monitor)** is a virtualization platform that allows multiple operating systems to run on a single host simultaneously within a full virtualization environment. A guest is an operating system (OS) that runs on a virtual machine in addition to the host or main OS.

With Red Hat Virtualization, each guests **memory** comes from a slice of the host's physical memory. For paravirtual guests, you can set both the initial memory and the maximum size of the virtual machine. You can add (or remove) physical memory to the virtual machine at runtime without exceeding the maximum size you specify. This process is called **ballooning**.

你可以用许多虚拟 **cpu**（称为 **vcpu**）来配置每个客户机。虚拟机管理程序按照物理 CPU 的负载来调度这些 **vcpu**。

你可以授予某个客户机任何数量的**虚拟磁盘 (virtual disks)**。客户机把这些磁盘视为硬盘或（对于完全虚拟化客户机而言）**CD-ROM** 设备。每个虚拟磁盘都以块设备或主机上的常规文件的方式供客户机使用。主机上的设备包含了客户机的完整的磁盘映像，通常也包括了分区表、多重分区和可能的 **LVM** 物理卷。

虚拟网络接口 (Virtual networking interface) 运行在客户机上。象虚拟以太网卡 (**VNIC**) 一样，其他接口可以运行在客户机上。这些网络接口用永久的虚拟 **MAC** 地址配置。新安装的客户机会缺省安装 **VNIC**，它的 **MAC** 地址从一个有着二千六百万地址的储备池里随机选取，所以两个客户机接受相同 **MAC** 地址是不太可能的。有着大量客户机的复杂站点可以手工分配 **MAC** 地址来确保在网络里的唯一性。

每个客户机都有一个连接至主机的**虚拟文本控制台 (text console)**。你可以把客户登录和控制台输出重定向到这个文本控制台。

你可以配置任何客户机来使用**虚拟图形控制台 (graphical console)**，这对应于物理主机上的普通的显示控制台。对于完全和半虚拟化客户机你都可以使用它。它具有标准的图形适配器的特征，如引导信息、图形化引导、多个虚拟终端以及 **X** 窗口系统。你也可以使用图形化键盘来配置虚拟键盘和鼠标。

Guests can be identified in any of three **identities**: domain name (domain-name), identity (domain-id), or **UUID**. The domain-name is a text string that corresponds to a guest configuration file. The domain-name is used to launch the guests, and when the guest runs the same name is used to identify and control it. The domain-id is a unique, non-persistent number that gets assigned to an active domain and is used to identify and control it. The **UUID** is a persistent, unique identifier that is controlled from the guest's configuration file and ensures that the guest is identified over time by system management tools. It is visible to the guest when it runs. A new **UUID** is automatically assigned to each guest by the system tools when the guest first installs.

第 2 章 操作系统支持

Red Hat Virtualization's paravirtualization mode allows you to utilize high performance virtualization on architectures that are potentially difficult to virtualize such as x86 based systems. To deploy paravirtualization across your operating system(s), you need access to the paravirtual guest kernels that are available from a respective Red Hat distro (for example, RHEL 4.0, RHEL 5.0, etc.). Whilst your operating system kernels must support Red Hat Virtualization, it is not necessary to modify user applications or libraries.

如果你有 Intel VT 和 AMD SVM CPU 硬件，Red Hat 虚拟化系统允许你运行未经修改的客户机内核。在 Intel VT 或 AMD SVM 系统里，你不必要移植你的操作系统来部署这个体系结构。Red Hat 虚拟化系统支持：

- 用于完全和半虚拟化的 Intel VT-x 或 AMD-V Pacifica 和 Vanderpool 技术。
- 用于 ia64 的 Intel VT-i
- Linux 和 UNIX 操作系统，包括 NetBSD、FreeBSD 和 Solaris。
- Microsoft Windows as an unmodified guest operating system with Intel Vanderpool or AMD's Pacifica technology.

要在 Hardware-assisted Virtual Machine (HVM)、Intel 或 AMD 平台的系统里运行完全虚拟化环境，你必须检查你的 CPU，确保它具备所需的能力。

要检查是否用于 Intel 支持的 CPU flags，键入下面的命令：

```
grep vmx /proc/cpuinfo
```

输出如下：

```
flags      :  fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush dts
acpi mmx fxsr sse sse2 ss ht tm syscall nx lm constant_tsc pni monitor
ds_cpl vmx est tm2 cx16 xtpr lahf_lm
```

如果出现了 vmx flag，那么你的 CPU 带有 Intel 支持。

要检查你是否有 AMD 支持的 CPU flag，键入下面的命令：

```
grep svm /proc/cpuinfo
cat /proc/cpuinfo | grep svm
```

输出如下：

```
flags      :  fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush dt
acpi mmx fxsr sse sse2 ss ht tm syscall nx mmtxt fxsr_opt rdtscp lm
3dnowext pni cx16 lahf_lm cmp_legacy svm cr8_legacy
```

如果出现了 svm flag，则表示你的 CPU 带有 AMD 支持。



注意

除了检查 CPU flag，你应该在系统的 BIOS 启用完全虚拟化。

第 3 章 硬件支持

Red Hat 虚拟化系统支持多处理器系统并允许你在 x86 体系结构的系统上用 P6（或更早）处理器运行 Red Hat 虚拟化系统：

- Celeron
- Pentium II
- Pentium III
- Pentium IV
- Xeon
- AMD Athlon
- AMD Duron

在红帽虚拟化系统里，32 位主机只运行 32 位半虚拟化客户机。64 位主机只运行 64 位半虚拟化客户机。64 位完全虚拟化主机可以运行 32 位、32 位 PAE 或 64 位客户机。32 位完全虚拟化主机可以运行 PAE 和非 PAE 完全虚拟化客户机。

对于 x86_64 系统，红帽企业 Linux 虚拟化系统内核不支持超过 32GB 的内存。如果你需要在超过 32GB 物理内存的系统上引导虚拟化内核，你必须在内核命令行后附加 `mem=32G` 参数。这个例子演示了在 `grub.conf` 文件里怎样采用正确的参数：

```
title Red Hat Enterprise Linux Server (2.6.18-4.elxen)
root (hd0, 0)
kernel /xen.gz-2.6.18-4-el5 mem=32G
module /vmlinuz -2.6.18-4.el5xen ro root=LABEL=/
module /initrd-2.6.18-4.el5xen.img
```

PAE (Physical Address Extension) 是一种为用户的应用程序增加物理或虚拟内存的技术。红帽虚拟化系统要求系统里有活动的 PAE。带有 PAE 支持的红帽虚拟化 32 位体系结构支持最高为 16GB 的物理内存。我们推荐你给系统里的每个客户机至少准备 256M 内存。x86/64 位机器里的红帽虚拟化系统最多可以寻址达 64GB 物理内存。红帽虚拟化内核在非 PAE 系统里无法运行。要知道你的系统是否支持 PAE，可以键入下面的命令：

```
grep pae /proc/cpuinfo
```

显示下面的输出：

```
flags : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 mmx fxsr sse
syscall mmtxt 3dnowext 3dnow up ts
```

如果你的输出和上面的相同（或近似），那么你的 CPU 支持 PAE。如果命令提示不显示任何输出，说明你的 CPU 不支持 PAE。

第 4 章 RED HAT 虚拟化系统的系统要求

下面列出的项目是 Red Hat 虚拟化系统所要求的：

- 正常运行的 Red Hat RHEL 5 Linux 系统
- 可用的 GRUB 引导装载程序
- 根访问权限
- P6 级（或更早）的处理器
- The Linux bridge-utils
- Linux 热插拔系统
- zlib 开发库
- Python 2.2 runtime
- initscripts

在安装过程中依赖关系被自动地配置。



注意

If your system CPU architecture is ia64, you need to manually install the `xen-ia64-guest-firmware` package to run a fully virtualized guest. This package is provided in the Supplementary CD and is not installed by default.

第 5 章 引导系统

在安装了Red Hat虚拟化组件后，你必须重新启动系统。当引导完成后，你必须象平常一样登录到系统里。而且在你启动Red Hat虚拟化系统之前，你必须以根用户登录。`xend` 守护进程应该已经被 `initscripts` 初始化，如果要手工启动 `xend`，可以输入：

```
service xend start
```

你也可以用 `chkconfig xend` 来使 `xend` 在引导时启动。

`xend` 节点控制守护进程执行和虚拟机相关的系统管理功能。这个守护进程控制虚拟化的资源，且和虚拟机进行交互。在你启动 `xend` 之前，你必须通过编辑 `xend` 的配置文件 `xend-config.xcp` 来指定操作参数，这个文件位于 `etc/xen` 目录下。

第 6 章 配置 GRUB

GNU Grand Unified Boot Loader (or GRUB) is a program which enables the user to select which installed operating system or kernel to load at system boot time. It also allows the user to pass arguments to the kernel. The GRUB configuration file (located in `/boot/grub/grub.conf`) is used to create a list of operating systems to boot in GRUB's menu interface. When you install the kernel-xen RPM, a post script adds kernel-xen entries to the GRUB configuration file. You can edit the `grub.conf` file and enable the following GRUB parameter:

```
title Red Hat Enterprise Linux Server (2.6.18-3.el5xen)
root    (hd0; 0)
kernel  /xen.gz.-2.6.18-3.el5
module  /vmlinuz-2.6..18-3.el5xen ro root=/dev/VolGroup00/LogVol100 rhgb
quiet
module  /initrd-2.6.18-3. el5xenxen.img
```

如果你设置与示例相同的 Linux grub 条目，引导装载程序会载入 `hypervisor`、`initrd` 映像和 Linux 内核。既然关于内核的条目是在所有其他条目之上，内核会先被载入内存。引导装载程序向 `hypervisor` 和 Linux 内核发送（和接受）命令行参数。下面的示例条目展示了怎样把 Domain0 linux 内核内存限制在 800MB 之内：

```
title Red Hat Enterprise Linux Server (2.6.18-3.el5xen)
root    (hd0; 0)
kernel  /xen.gz.-2.6.18-3.el5 dom0_mem=800M
module  /vmlinuz-2.6..18-3.el5xen ro root=/dev/VolGroup00/LogVol100 rhgb
quiet
module  /initrd-2.6.18-3. el5xenxen.img
```

你可以使用这些 GRUB 参数来配置虚拟化管理程序（`hypervisor`）：

```
mem
```

这限制了可用于 `domain0` 的内存数量。

```
com1=115200, 8n1
```

这启用了系统里的第一个串口来充当串行控制台（`com2` 被分配给下一个端口，诸如此类）。

```
dom0_mem
```

这限制了可用于 `domain0` 的内存数量。

```
dom0_max_vcpus
```

这限制了 `domain0` 可见的 CPU 数量。

acpi

这把 ACPI hypervisor 切换至 hypervisor 和 domain0。ACPI 参数的选项包括：

```
/* **** Linux config options: propagated to domain0 ****/  
/* "acpi=off":      Disables both ACPI table parsing and interpreter.  
*/  
/* "acpi=force":    Overrides the disable blacklist.  
*/  
/* "acpi=strict":   Disables out-of-spec workarounds.  
*/  
/* "acpi=ht":       Limits ACPI from boot-time to enable HT.  
*/  
/* "acpi=noirq":    Disables ACPI interrupt routing.  
*/
```

noacpi

这禁用了用于 interrupt delivery 的 ACPI。

第 7 章 引导客户机域

你可以使用 `xm` 程序来引导客户机域。你也可以用 `virsh` 和虚拟机管理者 (Virtual Machine Manager) 来引导客户机。引导客户机域的先决条件是先安装客户主机。下例使用了 `xm create` 子命令：

```
# xm create -c guestdomain1
```

`guestdomain1` 是你所引导的域的配置文件的名称。`-c` 选项表示在引导后连接到实际的控制台。

第 8 章 在引导时启动/停止域

你可以在任何时候启动或停止运行的域。Domain0 在重启之前等待所有运行域的关闭。你必须把你想关闭的域的配置文件的配置文件置于 `/etc/xen/` 目录下。所有你想在引导时启动的域必须有到 `/etc/xen/auto` 的符号链接。

```
chkconfig xendomains on
```

chkconfig xendomains on 命令不会自动地启动域，它将在下次引导时启动域。

```
chkconfig xendomains off
```

终止所有运行的Red Hat虚拟化域。**chkconfig xendomains off** 命令在下次引导时关闭所有的域。

第 9 章 配置文件

Red Hat Virtualization configuration files contain the following standard variables. Configuration items within these files must be enclosed in quotes (""). These configuration files reside in the `/etc/xen` directory.

表 9.1. 红帽虚拟化系统配置文件

项目	描述
<code>pae</code>	指定物理地址扩展配置数据。
<code>apic</code>	指定高级可编程的中断控制器配置数据。
<code>memory</code>	指定以 <code>megabyte</code> 为单位的内存大小
<code>vcpus</code>	指定虚拟 CPU 的数量。
<code>console</code>	指定导出域控制台的端口号。
<code>nic</code>	指定虚拟网络接口的数量。
<code>vif</code>	Lists the randomly-assigned MAC addresses and bridges assigned to use for the domain's network addresses.
<code>disk</code>	列出导出至域的块设备并以只读权限导出物理设备到域。
<code>dhcp</code>	用 DHCP 启用网络。
<code>netmask</code>	指定配置的 IP 掩码。
<code>gateway</code>	指定配置的 IP 网关。
<code>acpi</code>	指定高级配置电源接口的配置数据。

第 10 章 管理 CPU

Red Hat Virtualization allows a domain's virtual CPUs to associate with one or more host CPUs. This can be used to allocate real resources among one or more guests. This approach allows Red Hat Virtualization to make optimal use of processor resources when employing dual-core, hyperthreading, or other advanced CPU technologies. If you are running I/O intensive tasks, its typically better to dedicate either a hyperthread or entire core to run domain0. The Red Hat Virtualization credit scheduler automatically rebalances virtual cpus between physical ones, to maximize system use. The Red Hat Virtualization system allows the credit scheduler to move CPUs around as necessary, as long as the virtual CPU is pinned to a physical CPU.

第 11 章 移植域

移植是把运行的虚拟域从一台物理主机搬迁到另外一台主机。红帽虚拟化系统支持两种移植方式 – **offline** 和 **live**。**Offline** 移植通过暂停虚拟机、转移内存然后在目的主机上恢复执行的方式把虚拟机从一台主机移到另外一台主机。**Live** 移植做相同的事情，但不直接影响域。当执行 **live** 移植时，域继续它平常的活动，从用户的角度来看它并没有值得注意的地方。要初始化一个 **live** 移植，两台主机都必须运行红帽虚拟化系统以及 **xend** 守护进程。目的主机必须有足够的资源（如内存）来容纳移植后的域带宽（**bandwidth**）。源主机和目的主机都必须有相同的体系结构和虚拟化扩展（如 **i386-VT**、**x86-64-VT**、**x86-64-SVM** 等）且必须都位于相同的 L2 子网。

When a domain migrates its MAC and IP addresses move with it. Only virtual machines with the same layer-2 network and subnets will successfully migrate. If the destination node is on a different subnet, the administrator must manually configure a suitable EtherIP or IP tunnel in the remote node of domain0. The xend daemon stops the domain and copies the job over to the new node and restarts it. The Red Hat Virtualization RPM does not enable migration from any other host except the localhost (see the `/etc/xend-config.sxp` file for information). To allow the migration target to accept incoming migration requests from remote hosts, you must modify the target's `xen-relocation-hosts-allow` parameter. Be sure to carefully restrict which hosts are allowed to migrate, since there is no authentication.

既然这些域涉及如此大的文件分配，这个过程可能会消耗较久的时间。如果你移植带有开放网络连接的域，这些连接将在目的主机上保留，**SSH** 连接应该仍然可用。缺省的 Red Hat 虚拟化系统的 **iptables** 规则将不允许传入的移植连接。如要允许这种连接，你必须创建显性的 **iptables** 规则。

你可以使用 `xm migrate` 命令来执行 **offline** 移植：

```
xm migrate domain-id [destination domain]
```

你可以使用 `xm migrate` 命令来执行 **live** 移植：

```
xm migrate domain-id -l [destination domain]
```

You may need to reconnect to the domain's console on the new machine. You can use the `xm console` command to reconnect.

第 12 章 配置网络应用

把红帽虚拟化系统集成到你的网络体系里是一个复杂的过程，且依赖于基础架构，它可能需要自定义的配置来部署多个以太网接口和设置桥接。

Each domain network interface is connected to a virtual network interface in `dom0` by a point to point link. These devices are `vif <domid>` and `<vifid>.vif1.0` for the first interface in domain 1; `vif3.1` for the second interface in domain 3.

Domain0 处理这些虚拟接口上的通信，对于网桥、路由、速率限制等，它采用标准的 Linux 传统。`xend` 守护进程使用两个 shell 脚本来执行网络和新的虚拟接口的初始配置。这些脚本为所有的虚拟接口配置单一的网桥。你可以自定义这些脚本来配置其他的路由和网桥。

Red Hat Virtualization's virtual networking is controlled by the two shell scripts, `network-bridge` and `vif-bridge`. `xend` calls these scripts when certain events occur. Arguments can be passed to the scripts to provide additional contextual information. These scripts are located in the `/etc/xen/scripts` directory. You can change script properties by modifying the `xend-config.sxp` configuration file located in the `/etc/xen` directory.

network-bridge – 当 `xend` 启动或停止时，这个脚本初始化或者停止虚拟网络。之后，初始化过程创建网桥 `xen-br0` 并把 `eth0` 移至这个网桥上，且修改对应的路由。当 `xend` 最后退出时，它删除这个网桥和 `eth0`，由此恢复原始的 IP 和路由配置。

vif-bridge - 对于域里的每个虚拟接口，这个脚本都被调用。它配置防火墙规则且能够把 `vif` 加入到合适的网桥上。

你可以使用其他脚本在你的网络里设立 Red Hat 虚拟化系统，如 `network-route`、`network-nat`、`vif-route`，和 `vif-nat`。或者这些脚本可以用自定义的变量来替代。

第 13 章 设置 DOMAIN0 的安全性

When deploying Red Hat Virtualization on your corporate infrastructure, you must ensure that domain0 cannot be compromised. Domain0 is the privileged domain that handles system management. If domain0 is insecure, all other domains in the system are vulnerable. There are several ways to implement security you should know about when integrating Red Hat Virtualization into your systems. Together with other people in your organization, you should create a 'deployment plan' that contains the operating specifications and services that will run on Red Hat Virtualization, and what is needed to support these services. Here are some security issues to consider when putting together a deployment plan:

- 只运行最小数目的必需的服务。不要在 domain0 里运行太多的任务和服务。运行的服务越少，安全性越高。
- 启用 SeLINUX 帮助提高 domain0 的安全性。
- 使用防火墙来限制到 domain0 的通信量。你可以设置采用 default-reject 规则的防火墙，这将有助于避免对 domain0 的攻击。限制网络 facing 服务也是很重要的。
- 不要允许普通用户访问 domain0。如果你允许普通用户访问 domain0，这可能会导致 domain0 易受攻击。记住，domain0 是专用的，允许非专用帐号的访问可能会降低安全级别。

第 14 章 存储

有几个方法来管理虚拟机的存储。你可以把 `domain0` 的物理块设备（硬盘或分区）作为虚拟块设备（`virtual block device`，`VBD`）导出到客户机域。你也可以直接从分区映像里导出为 `file-backed` 虚拟块设备。在安装过程中，红帽虚拟化系统缺省启用 `LVM` 和 `blktap`。你也可以使用标准的网络协议，如 `NFS`、`CLVM` 或 `iSCSI` 来提供虚拟机的存储。

第 15 章 用 VIRSH 管理虚拟机

你可以使用 **virsh** 程序来管理虚拟机。这个工具是用 **libvirt management API** 构建并作为 **xm** 工具或图形化虚拟机管理者的替代来使用的。**Unprivileged** 用户可以把这个工具用于只读操作。如果你打算运行 **xend/qemu**，你应该把 **xend/qemu** 作为服务来运行。在修改了相应的配置文件后，重新启动系统，**xend/qemu** 将作为服务运行。你可以把 **virsh** 用在脚本里。和 **xm** 工具一样，你可以在命令行里运行 **virsh**。

15.1. 连接至监控程序

你可以使用 **virsh** 来初始化一个监控程序会话：

```
virsh connect <name>
```

Where **<name>** is the machine name of the hypervisor. If you want to initiate a read-only connection, append the above command with **-readonly**.

15.2. 创建虚拟机

你可以从 **XML** 机器定义里创建一个新的虚拟机会话。如果你有一个现存的用 **xm** 创建的客户端，你也可以为它创建一个虚拟机：

```
virsh create <path to XML configuration file>
```

15.3. 配置 XML 转储

你可以使用 **virsh** 为现有的虚拟机执行一个数据转储。

```
virsh dumpxml [domain-id | domain-name | domain-uuid]
```

This command outputs the domain information (in XML) to **stdout**. If you save the data to a file, you can use the **create** option to recreate the virtual machine.

15.4. 挂起虚拟机

你可以使用 **virsh** 来挂起某个域：

```
virsh suspend [domain-id | domain-name | domain-uuid]
```

When a domain is in a suspended state, it still consumes system RAM. There will also be no disk or network I/O when suspended. This operation is immediate and the virtual machine must be restarted with the **resume** option.

15.5. 恢复虚拟机的运行

你可以用 **virsh** 来恢复一个挂起的虚拟机：

```
virsh resume [domain-id | domain-name | domain-uuid]
```

这个操作是立时生效的，虚拟机状态将处于 **suspend** 和 **resume** 的循环中。

15.6. 保存虚拟机

你可以使用 **virsh** 来把虚拟机的当前状态保存到一个文件里：

```
virsh save [domain-name][domain-id | domain-uuid][filename]
```

This stops the virtual machine you specify and saves the data to a file, which may take some time given the amount of memory in use by your virtual machine. You can restore the state of the virtual machine with the **restore** option.

15.7. 恢复虚拟机

You can use **virsh** to restore a virtual machine that you previously saved with the **virsh save** option:

```
virsh restore [filename]
```

This restarts the saved virtual machine, which may take some time. The virtual machine's name and UUID are preserved but are allocated for a new id.

15.8. 关闭虚拟机

你可以使用 **virsh** 来关闭虚拟机：

```
virsh shutdown [domain-id | domain-name | domain-uuid]
```

通过修改 **xmdomain.cfg** 文件的 **on_shutdown** 参数，你可以控制虚拟机重启的行为。

15.9. 重新启动虚拟机

你可以使用 **virsh** 来重启虚拟机：

```
virsh reboot [domain-id | domain-name | domain-uuid]
```

通过修改 **xmdomain.cfg** 文件的 **on_reboot** 参数，你可以控制虚拟机重启的行为。

15.10. 终止域

你可以用 **virsh** 来终止某个虚拟机：

```
virsh destroy [domain-name | domain-id | domain-uuid]
```

This command does an immediate ungraceful shutdown and stops any guest domain sessions (which could potentially lead to file corrupted filesystems still in use by the virtual machine). You should use the **destroy** option only when the virtual machine's operating system is non-responsive. For a paravirtualized virtual machine, you should use the **shutdown** option.

15.11. 把域名转换为 DOMAIN ID

你可以用 **virsh** 把把域名或 UUID 转换为 Domain ID。


```
virsh domid [domain-name | domain-uuid]
```

15.12. 把 DOMAIN ID 转换为域名

你可以用 **virsh** 把 Domain ID 或 UUID 转换为域名：

```
virsh domname [domain-name | domain-uuid]
```

15.13. 把域名转换为 UUID

你可以用 **virsh** 把域名转换为 UUID：

```
virsh domuuid [domain-id | domain-uuid]
```

15.14. 显示虚拟机信息

你可以用 **virsh** 来显示以 domain ID、域名或 UUID 识别的虚拟机的信息：

```
virsh dominfo [domain-id | domain-name | domain-uuid]
```

15.15. 显示节点信息

你可以使用 **virsh** 来显示节点信息：

```
virsh nodeinfo
```

输出和下面类似：

```
CPU model           x86_64
CPU (s)             8
CPU frequency       2895 Mhz
CPU socket(s)       2
Core(s) per socket  2
Threads per core:   2
Numa cell(s)        1
Memory size:        1046528 kb
```

这显示了节点信息和支持虚拟化进程的机器。

15.16. 显示虚拟机信息

你可以用 **virsh** 来显示虚拟机列表和当前状态：

```
virsh list domain-name [ --inactive | --all]
```

--inactive 选项列出不活动的域（已经被定义但当前不活动的域）。**--all** 选项列出所有的域，包括活动和不活动的。输出应该和下例中的类似：

```
ID                Name                State
```

```
0          Domain0          running
1          Domain202        paused
2          Domain010        inactive
3          Domain9600        crashed
```

这里有六个域状态：

```
running  lists domains currently active on the CPU
blocked  lists domains that are blocked
paused   lists domains that are suspended
shutdown lists domains that are in process of shutting down
shutoff  lists domains that are completely down.
crashed  lists domains that are crashed
```

15.17. 显示虚拟 CPU 信息

你可以用 **virsh** 来显示虚拟机里的虚拟 CPU 信息：

```
virsh vcpuinfo [domain-id | domain-name | domain-uuid]
```

15.18. 配置虚拟 CPU 关联

你可以用 **virsh** 来配置虚拟 CPU 和物理 CPU 的关系：

```
virsh vcpupin [domain-id | domain-name | domain-uuid] [vcpu] , [cpulist]
```

这里的 **[vcpu]** 是虚拟 VCPU 的号码而 **[cpulist]** 列出了物理 CPU 序号。

15.19. 配置虚拟 CPU 的数量

You can use **virsh** to modify a Virtual Machine's number of CPUs:

```
virsh setvcpus [domain-name | domain-id | domain-uuid] [count]
```

注意，新的数量不能超过你在创建虚拟机时指定的数量。

15.20. 配置内存分配

You can use **virsh** to modify a domain's memory allocation:

```
virsh setmem [domain-id | domain-name] [count]
```

You must specify the `[count]` in kilobytes. Note that the new count cannot exceed the amount you specified when you created the Virtual Machine. Values lower than 64 MB probably won't work. You can adjust the Virtual Machine memory as necessary.

15.21. 配置最大内存数量

You can use `virsh` to modify a Virtual Machine's maximum memory:

```
virsh setmaxmem [domain-name | domain-id | domain-uuid] [count]
```

You must specify the `[count]` in kilobytes. Note that the new count cannot exceed the amount you specified when you created the Virtual Machine. Values lower than 64 MB probably won't work. The maximum memory doesn't affect the current use of the Virtual Machine (unless the new value is lower which should shrink memory usage).

15.22. MANAGING VIRTUAL NETWORKS

You can use `virsh` to manage virtual networks. To list virtual networks:

```
virsh net-list
```

This command generates output similar to:

```
[root@domain ~]# virsh net-list
Name                State      Autostart
-----
default             active    yes
vnet1               active    yes
vnet2               active    yes
```

To view network information for a specific virtual network:

```
virsh net-dumpxml [vnet name]
```

This displays information about a specified virtual network in XML format:

```
[root@domain ~]# virsh net-dumpxml vnet1
<network>
  <name>vnet1</name>
  <uuid>98361b46-1581-acb7-1643-85a412626e70</uuid>
  <forward dev='eth0' />
  <bridge name='vnet0' stp='on' forwardDelay='0' />
  <ip address='192.168.100.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.100.128' end='192.168.100.254' />
    </dhcp>
  </ip>
</network>
```

Other `virsh` commands used in managing virtual networks are:

- **virsh net-autostart *[network name]*** – Autostart a network specified as *[network name]*
- **virsh net-create *[XML file]*** – Generates and starts a new network using a preexisting XML file
- **virsh net-define *[XML file]*** – Generates a new network from a preexisting XML file without starting it
- **virsh net-destroy *[network name]*** – Destroy a network specified as *[network name]*
- **virsh net-name *[network UUID]*** – Convert a specified *[network UUID]* to a network name
- **virsh net-uuid *[network name]*** – Convert a specified *[network name]* to a network UUID
- **virsh net-start *[name of an inactive network]*** – Starts a previously undefined inactive network
- **virsh net-undefine *[name of an inactive network]*** – Undefine an inactive network

第 16 章 用 XEND 管理虚拟机

xend 节点控制守护进程执行某些与虚拟机相关的系统管理功能。这个守护进程控制虚拟化的资源，而且 **xend** 必须与虚拟机进行交互。在你启动 **xend** 之前，你必须通过编辑 **xend** 的配置文件 **xend-config.sxp** 来指定操作参数，这个文件位于 **etc/xen** 目录。下面是你可以在 **xend-config.sxp** 配置文件里启用或禁止的参数：

表 16.1. 红帽虚拟化系统的 **xend** 配置参数

项目	描述
console-limit	Determines the console server's memory buffer limit and assigns values on a per-domain basis
min-mem	给 domain0 保留的最小内存数量（以 MB 为单位），如果为 0，则值不变化。
dom0 cpus	指定 domain0 使用的 CPU 数量（缺省情况下至少分配一个 CPU）
enable-dump	指定当发生崩溃时是否启用转储（缺省为 0）
external-migration-tool	指定用来处理外部设备移植的脚本或应用程序（脚本必须位于 etc/xen/scripts/external-device-migrate ）
logfile	指定日志文件的位置（缺省为 /var/log/xend.log ）
loglevel	指定日志模式：DEBUG、INFO、WARNING、ERROR 或 CRITICAL（缺省为 DEBUG）
network-script	指定启用网络环境的脚本（脚本必须位于 etc/xen/scripts 目录）
xend-http-server	是否启用 http stream 数据包管理服务器（缺省为 no）
xend-unix-server	启用 unix 域套接字服务器（套接字服务器是一个通信终点，它处理底层的网络连接并接受或拒绝转入的连接）。
xend-relocation-server	启用用于跨机器移植的 relocation 服务器（缺省为 no）
xend-unix-path	指定 xend-unix-server 命令输出数据的位置（缺省是 var/lib/xend/xend-socket ）
xend-port	指定 http 管理服务器使用的端口（缺省为 8000）
xend-relocation-port	指定 relocation 服务器所使用的端口（缺省为 8002）

项目	描述
<code>xend-relocation-address</code>	指定虚拟机允许系统移植的地址
<code>xend-address</code>	指定域套接字服务器绑定的地址

在设置了这些操作参数后，你应该校验 `xend` 是否正在运行，如果没有，就初始化它。在命令提示行下，你可以用下面的命令启动 `xend` 守护进程。

```
service xend start
```

你可以用 `xend` 来停止这个守护进程：

```
service xend stop
```

这个命令停止了守护进程。

你也可以用 `xend` 来重新启动守护进程：

```
service xend restart
```

守护进程再次启动了。

你可以检查 `xend` 守护进程的状态。

```
service xend status
```

The output displays the daemon's status.

第 17 章 用虚拟机管理者 (VIRTUAL MACHINE MANAGER) 管理虚拟机

本部分内容描述了 Red Hat 虚拟机管理者 (VMM) 窗口、对话框和不同的图形界面控制。

17.1. 虚拟机管理者架构

红帽虚拟化系统是一个软件组件的集合，它们组合在一起对虚拟机进行管理。虚拟机管理者 (VMM) 提供了系统里的虚拟机的图形化视图。你可以用 VMM 来定义并行和完全虚拟化机器。使用虚拟机管理者，你可以执行任何虚拟化管理任务，包括分配内存、分配虚拟 CPU、监控操作性能以及保存、恢复、暂停、继续执行和关闭虚拟系统。它也允许你访问文本和图形化控制台。红帽虚拟化系统从底层的硬件和网络配置里抽象出 CPU 和内存资源。这集中了处理资源，且可以动态地分配给应用程序和服务。芯片级的虚拟化使基于 Intel VT 和 AMD Pacifica 硬件的操作系统在监控程序里运行。

17.2. OPEN CONNECTION 窗口

This window appears first and prompts the user to choose a hypervisor session. Non-privileged users can initiate a read-only session. Root users can start a session with full blown read-write status. For normal use, select the **Local Xen host** option. You start the Virtual Machine Manager test mode by selecting the **Other hypervisor** and then type `test:///default` in the URL field beneath. Once in test mode, you can connect to a libvirt dummy hypervisor. Note that although the **Remote Xen host** screen is visible, the functionality to connect to such a host is not implemented into Red Hat Enterprise Linux 5.1.



图 17.1. Virtual Machine Manager Connection window

17.3. 虚拟机管理者窗口

这个主窗口显示了所有运行的虚拟机和分配给它们的资源（包括 domain0），你可以决定显示哪些字段。双击某个虚拟机将出现这个虚拟机的控制台。选择一个虚拟机并双击 **Display** 按钮，将出现这个机器的 **Details** 窗口。你也可以访问 **File** 菜单来创建新的虚拟机。

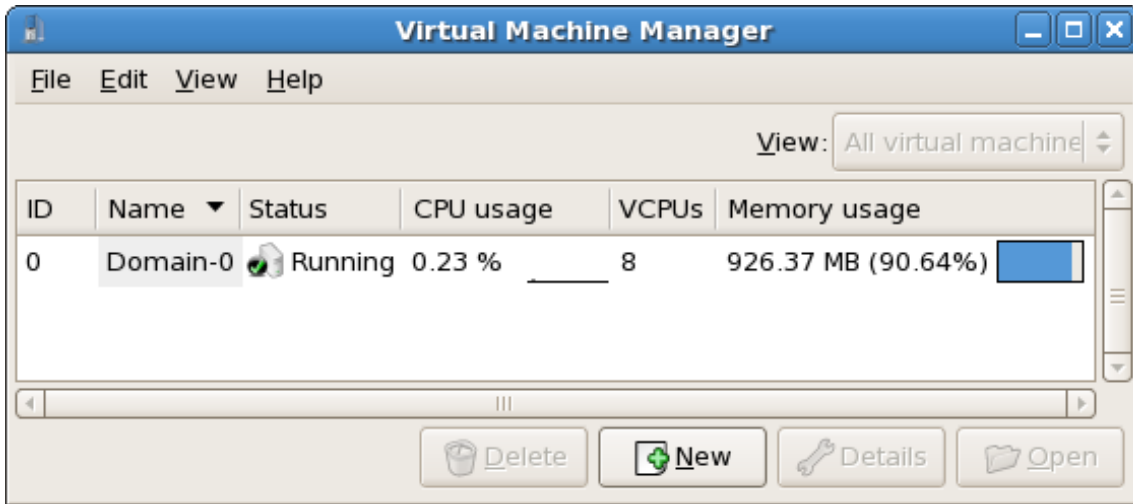


图 17.2. Virtual Machine Manager main window

17.4. 虚拟机 DETAILS 窗口

This window displays graphs and statistics of a guest's live resource utilization data available from the Red Hat Virtualization Virtual Machine Manager. The UUID field displays the globally unique identifier for the virtual machines(s).

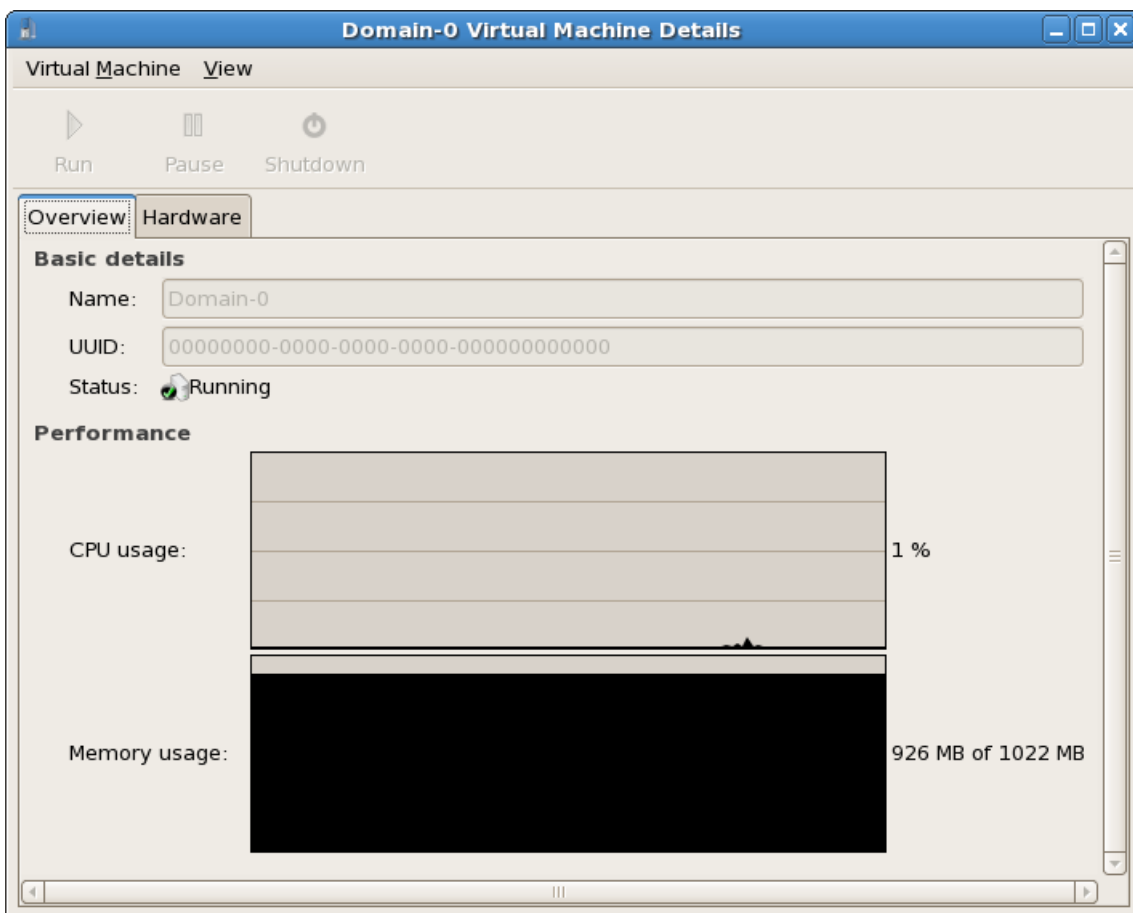


图 17.3. Virtual Machine Manager Details window

17.5. 虚拟机图形化控制台

This window displays a virtual machine's graphical console. Paravirtual and full virtual machines use different techniques to export their local virtual framebuffers, but both technologies use VNC to make

them available to the Virtual Machine Manager's console window. If your virtual machine is set to require authentication, the Virtual Machine Graphical console prompts you for a password before the display appears.

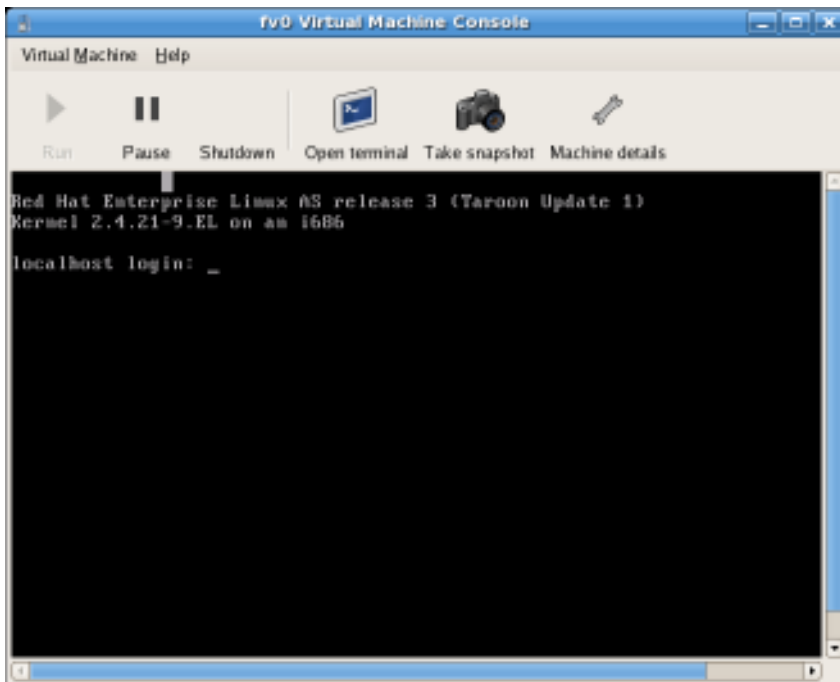


图 17.4. 图形化控制台

Your local desktop can intercept key combinations (for example, Ctrl+Alt+F11) to prevent them from being sent to the guest machine. You can use the Virtual Machine Manager's 'sticky key' capability to send these sequences. You must press any modifier key (like Ctrl or Alt) 3 times and the key you specify gets treated as active until the next non-modifier key is pressed. Then you can send Ctrl-Alt-F11 to the guest by entering the key sequence 'Ctrl Ctrl Ctrl Alt+F1'.

17.6. STARTING THE VIRTUAL MACHINE MANAGER

要启动虚拟机管理者会话，在 **Applications** 菜单，点击 **System Tools**，然后选择 **Virtual Machine Manager**。

虚拟机管理者主窗口将出现。

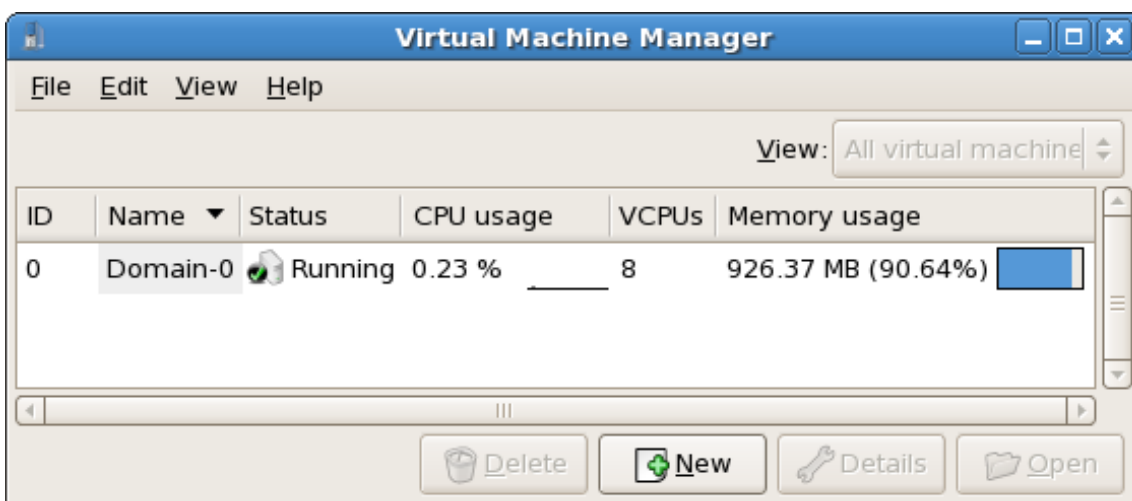


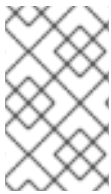
图 17.5. Starting the Virtual Machine Manager

17.7. 创建新的虚拟机

虚拟机管理者 (virt-manager) 是管理虚拟机的桌面程序。

You can use Red Hat's Virtual Machine Manager to:

- 创建新的域。
- Configure or adjust a domain's resource allocation and virtual hardware.
- 统计运行的域的性能和资源利用情况。
- 显示性能和资源利用情况的图表。
- 使用内嵌的 VNC 客户端浏览器来作为客户机域的完全图形化控制台。



注意

You must install Red Hat Enterprise Linux 5.1, virt-manager, and the kernel packages on all systems that require virtualization. All systems then must be booted and running the Red Hat Virtualization kernel.

下面是使用虚拟机监控程序在红帽企业 Linux 5 上安装客户机操作系统所需要的步骤：

过程 17.1. 创建客户机操作系统

1. 在 **Applications** 菜单，选择 **System Tools**，然后选择 **Virtual Machine Manager**。

虚拟机管理者主窗口将出现。

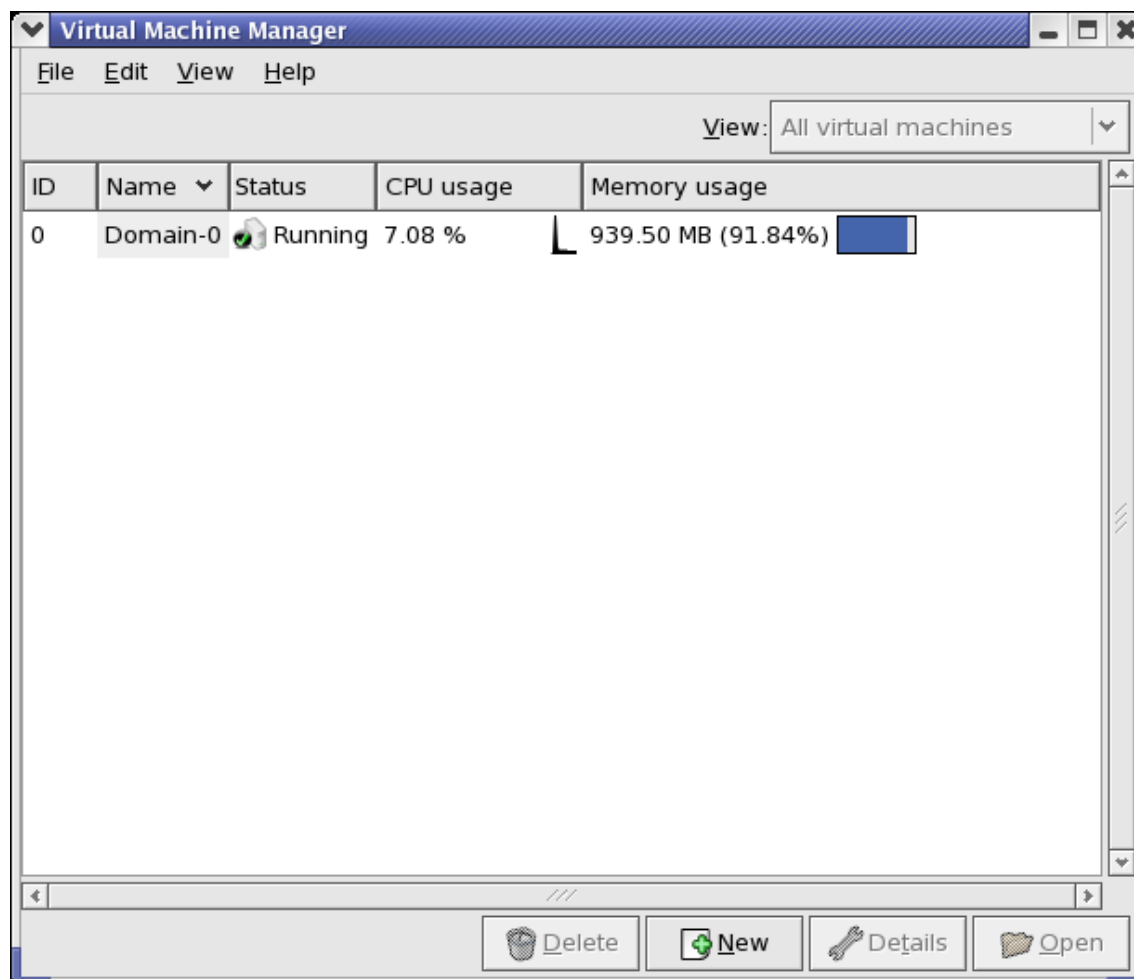


图 17.6. Virtual Machine Manager window

2. 在 **File** 菜单，选择 **New machine**。

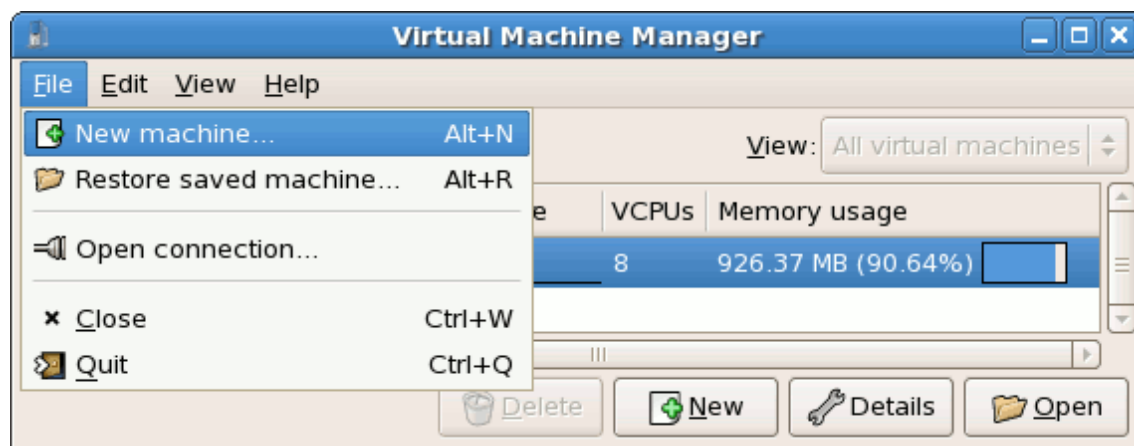


图 17.7. 选择一个新的机器

创建新虚拟机的向导将出现。

3. Click **Forward**.

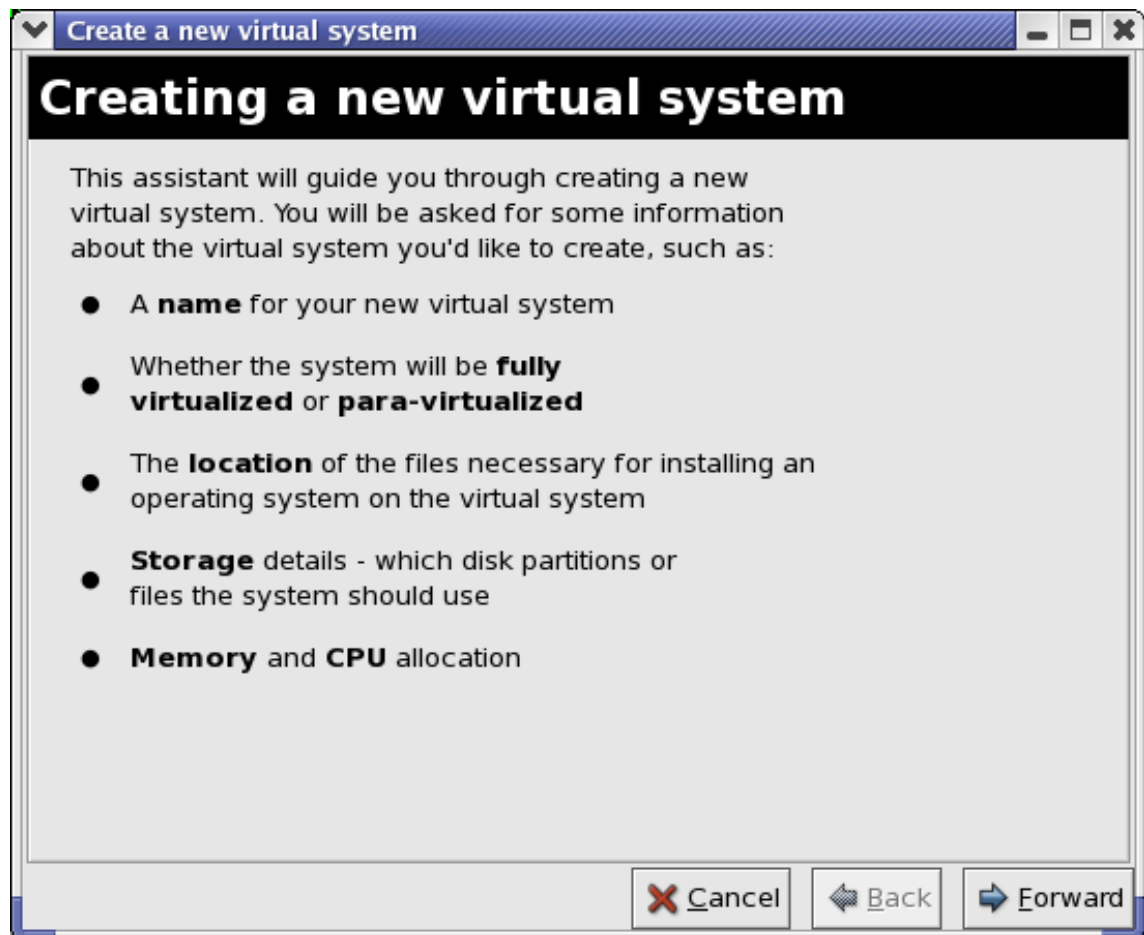


图 17.8. 创建新虚拟系统的向导

4. Enter the name of the new virtual system and then click **Forward**.

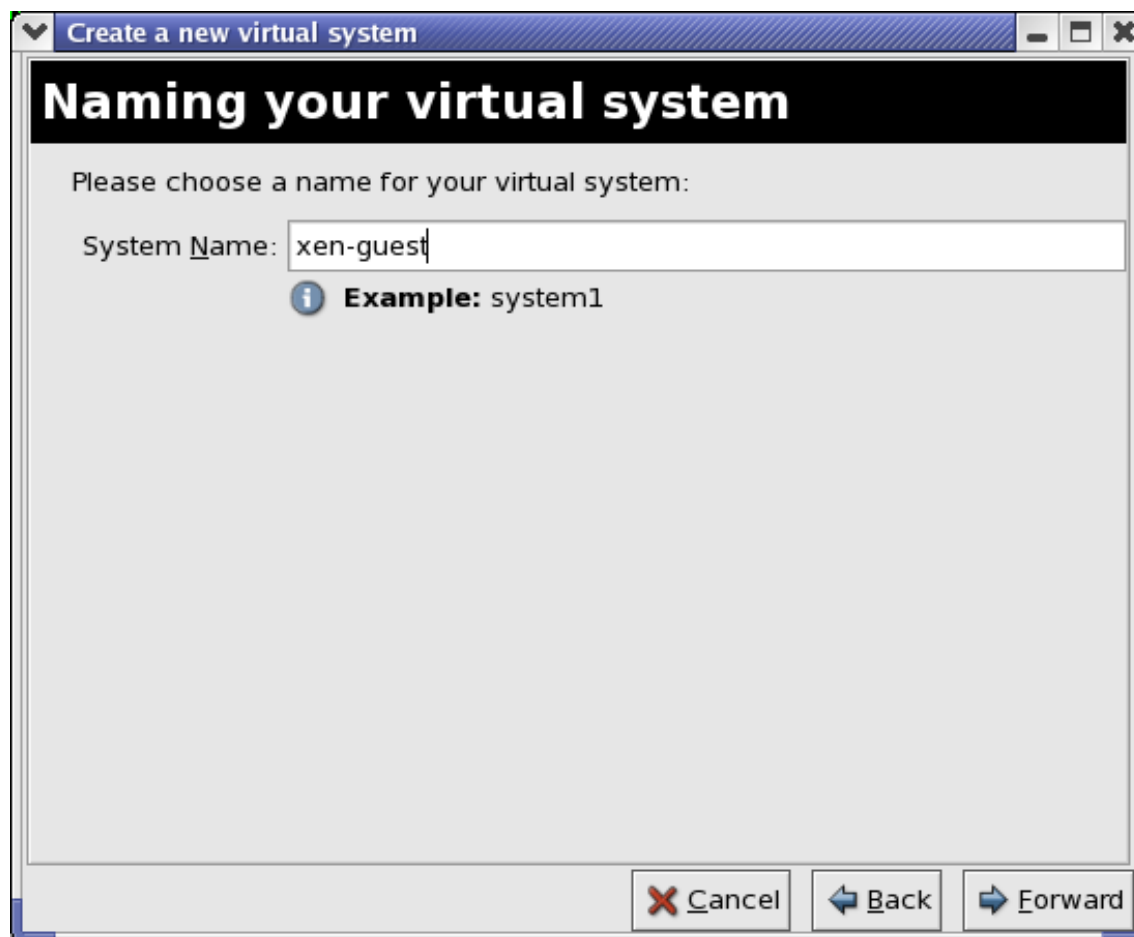


图 17.9. 命名虚拟系统

5. 输入安装介质的位置。kickstart 文件的位置是可选的。然后点击 **Forward** 。

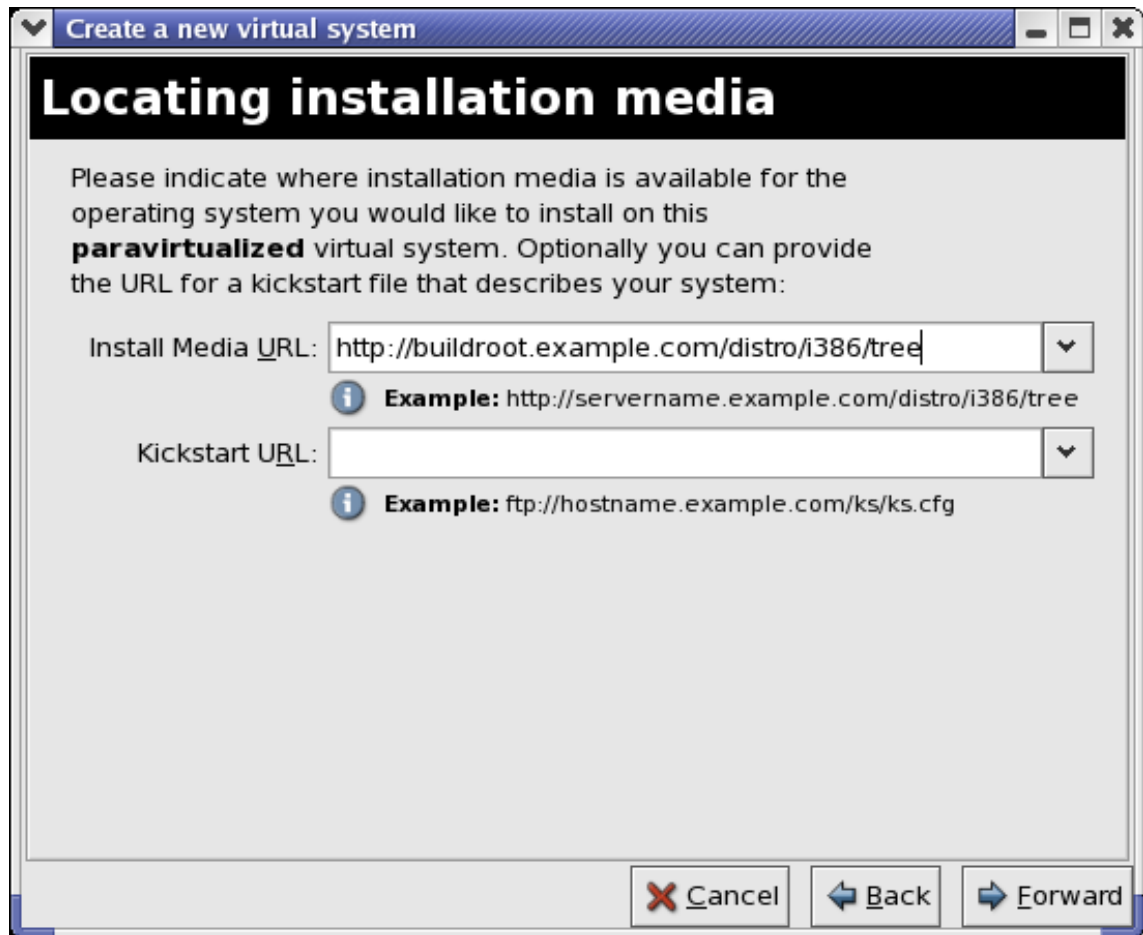
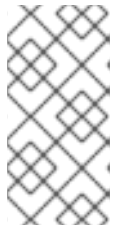


图 17.10. 定位安装介质

6. 安装至一个物理磁盘分区或是某个文件里的虚拟文件系统。



注意

这个例子在文件里安装了一个虚拟系统。

SELinux policy only allows xen disk images to reside in `/var/lib/xen/images`.

Open a terminal and create the `/xen` directory and set the SELinux policy with the command `restorecon -v /xen`. Specify your location and the size of the virtual disk, then click Forward.



图 17.11. 分配存储空间

7. Select memory to allocate the guest and the number of virtual CPUs then click **Forward**.

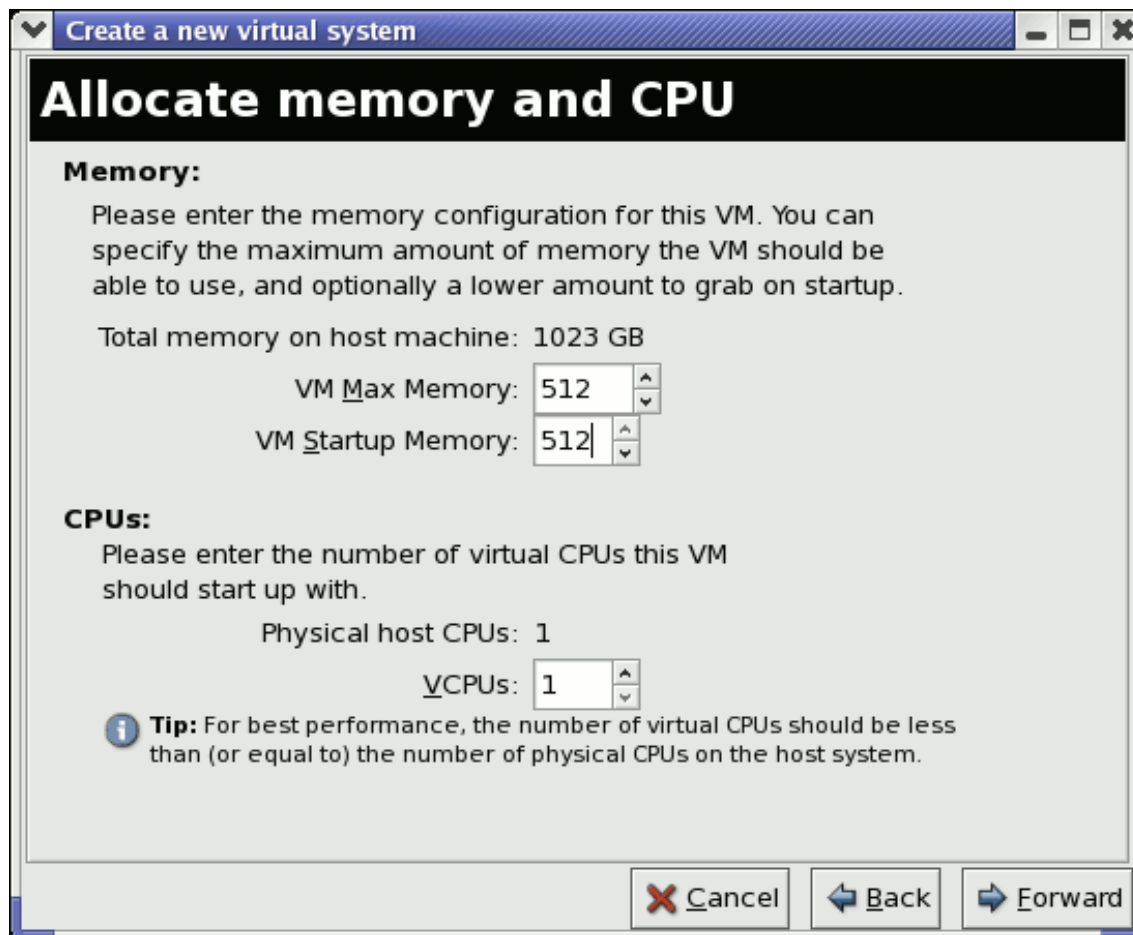


图 17.12. 分配内存和 CPU

8. Select **Forward** to open a console and the files start to install.

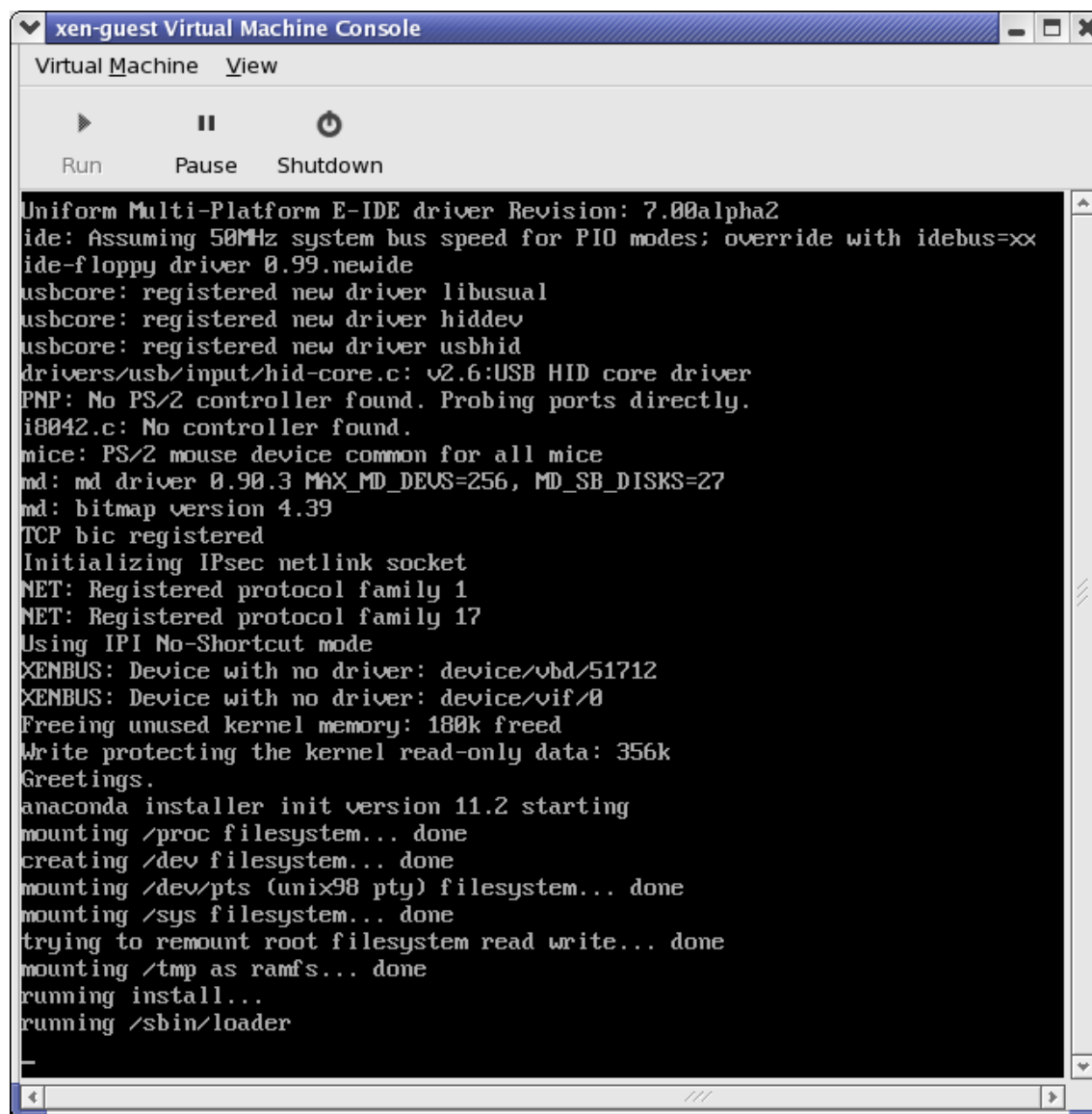


图 17.13. 分配内存和 CPU

9. 在这个窗口里完成安装。

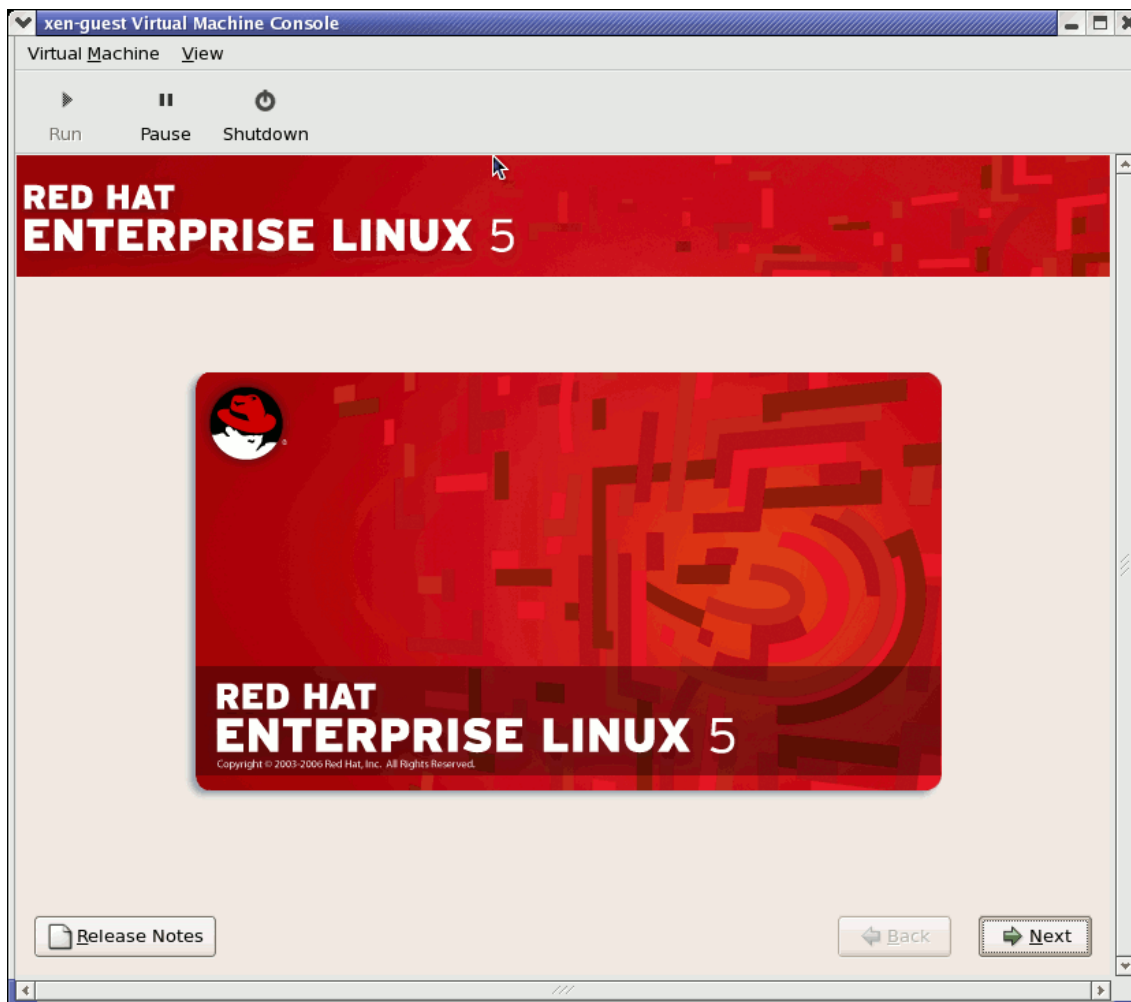


图 17.14. 安装开始...



警告

When installing Red Hat Enterprise Linux 5.1 on a fully virtualized guest, do *not* use the `kernel-xen` kernel. Using this kernel on fully virtualized guests can cause your system to hang.

If you are using an *Installation Number* when installing Red Hat Enterprise Linux 5.1 on a fully virtualized guest, be sure to deselect the **Virtualization** package group during the installation. The **Virtualization** package group option installs the `kernel-xen` kernel.

Note that paravirtualized guests are not affected by this issue. Paravirtualized guests always use the `kernel-xen` kernel.

10. Type `xm create -c xen-guest` to start the Red Hat Enterprise Linux 5.1 guest. Right click on the guest in the Virtual Machine Manager and choose **Open** to open a virtual console.

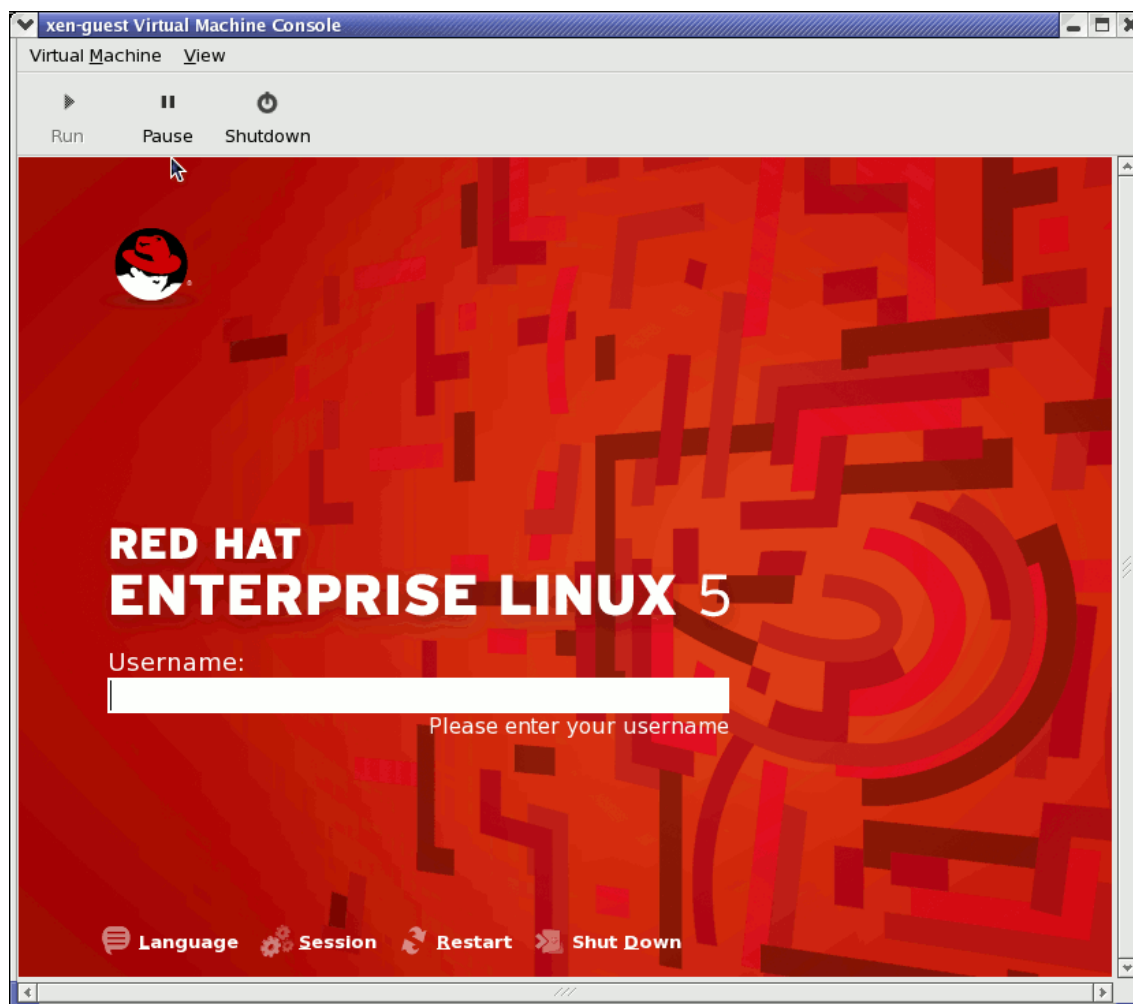


图 17.15. Red Hat Enterprise Linux 5.1 (guest)

11. 输入用户名和密码来继续使用虚拟机管理者。

17.8. 恢复保存的虚拟机

在启动虚拟机管理者后，系统里的所有虚拟机都显示在主窗口里。Domain 0 是你的主机系统。如果看不到虚拟机条目，说明系统里没有虚拟机在运行。

恢复以前保存的会话：

1. 在 **File** 菜单，选择 **Restore a saved machine**。

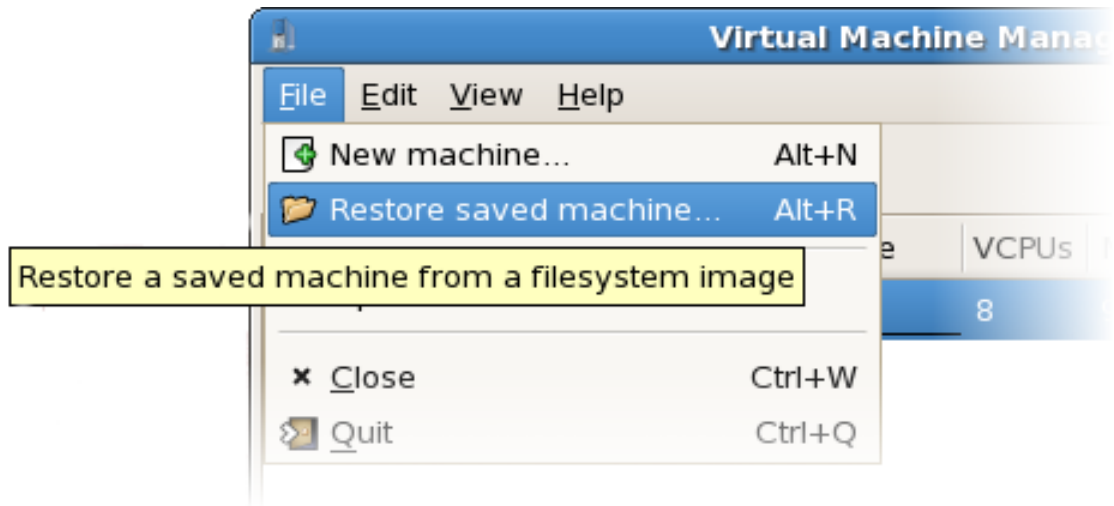


图 17.16. Restoring a Virtual Machine

2. 恢复虚拟机主窗口将出现。

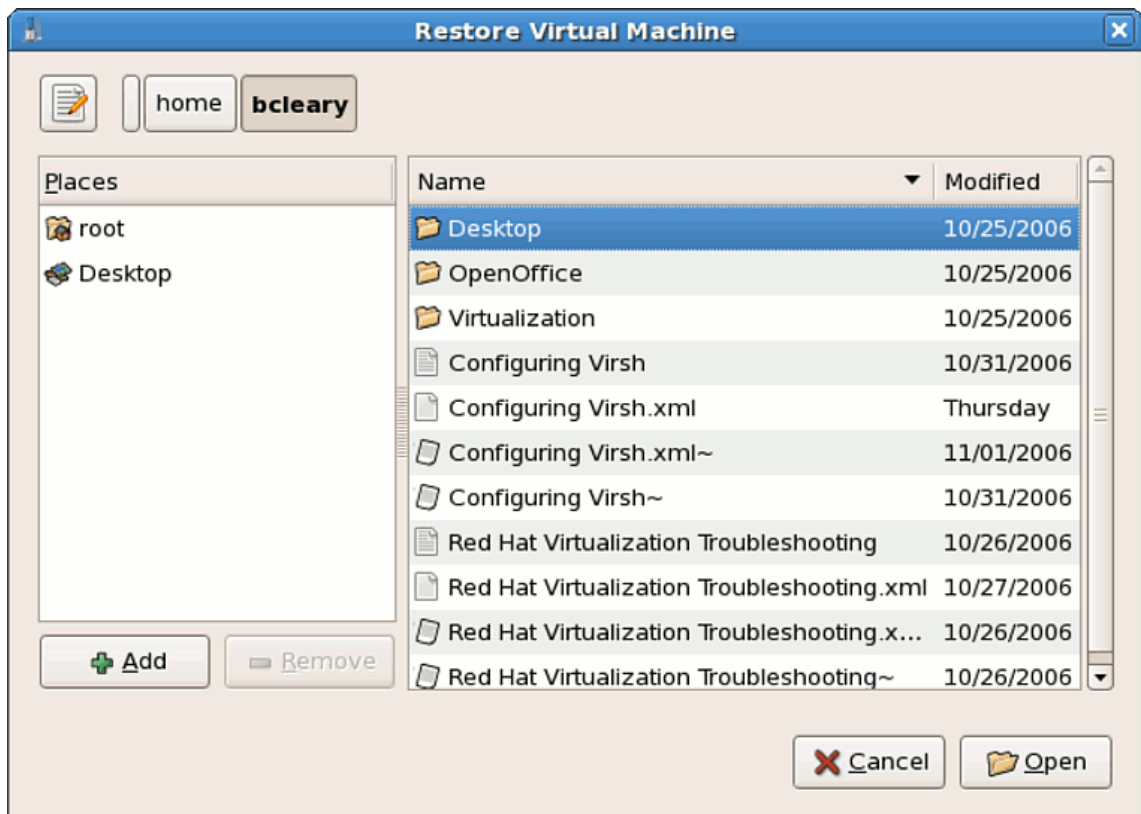


图 17.17. 选择保存的虚拟机会话

3. 导航至正确的目录并选择保存的会话文件。
4. 点击 **Open**。

保存的虚拟系统将出现在虚拟机管理者主窗口。

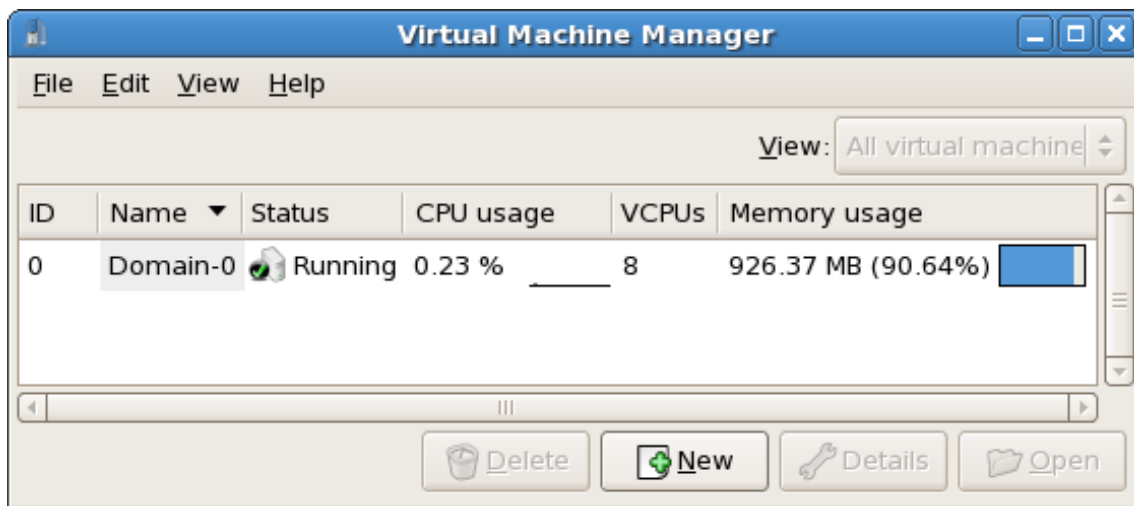


图 17.18. 被恢复的虚拟机管理者会话

17.9. DISPLAYING VIRTUAL MACHINE DETAILS

你可以使用虚拟机监控程序来查看系统里的任何虚拟机的活动数据。

To view a virtual system's details:

1. 在虚拟机管理者主窗口，高亮显示你要查看的虚拟机。

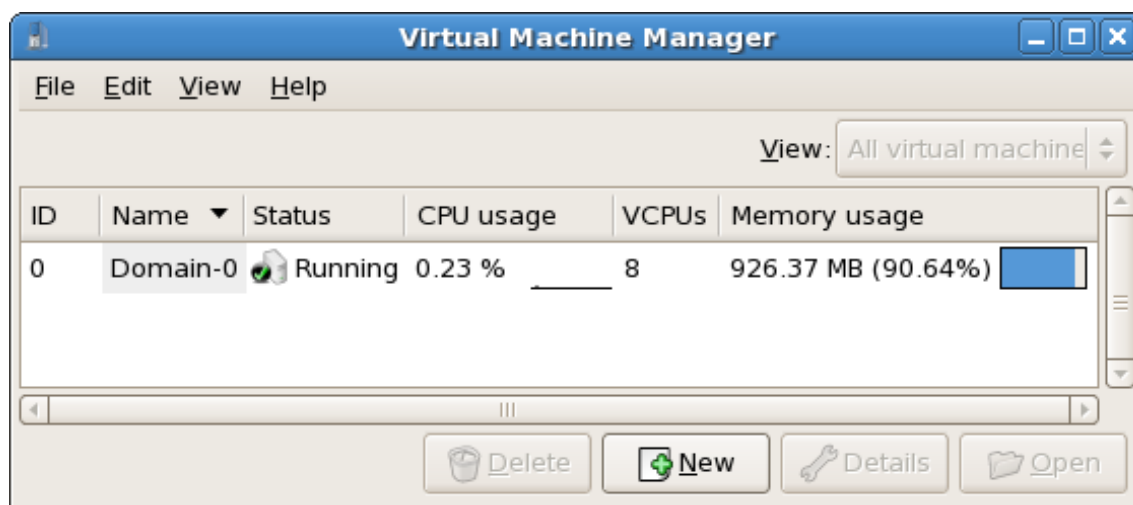


图 17.19. 选择要显示的虚拟机

2. 在虚拟机管理者的 **Edit** 菜单，选择 **Machine Details**（或在虚拟机管理者主窗口的底部点击 **Details** 按钮）。

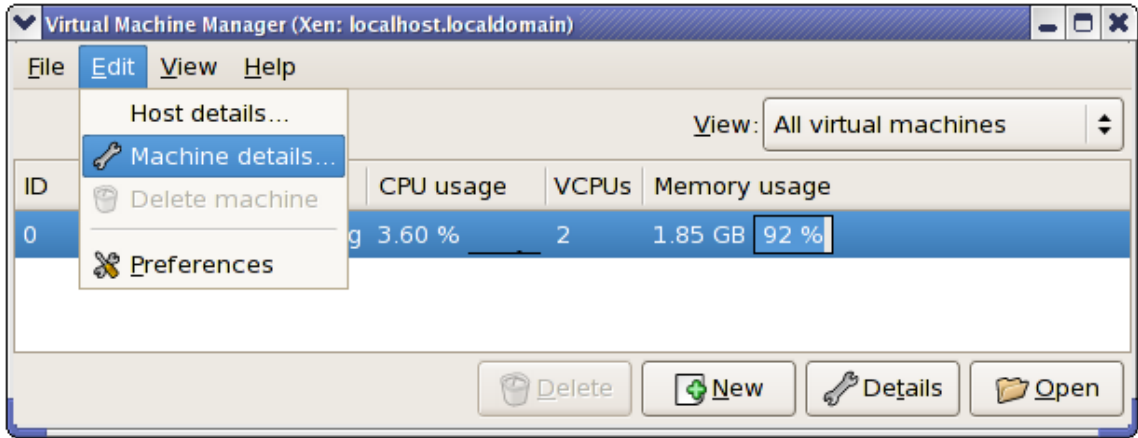


图 17.20. 显示虚拟机细节菜单

Virtual Machine Details Overview 窗口将出现。这个窗口显示了你指定的域的 CPU 和内存使用情况。

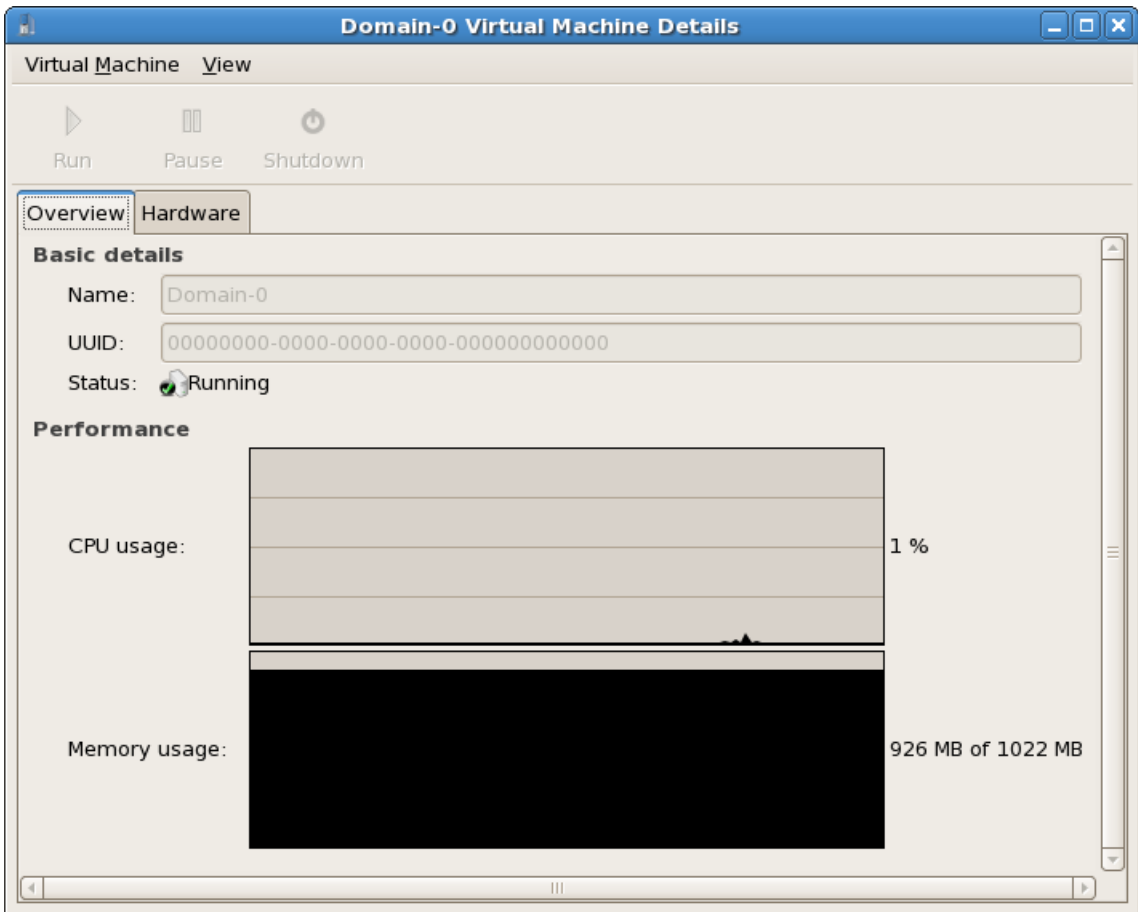


图 17.21. 显示 Virtual Machine Details Overview

- 3. 在 Virtual Machine Details 窗口，点击 **Hardware** 页。

Virtual Machine Details Hardware 窗口将出现。

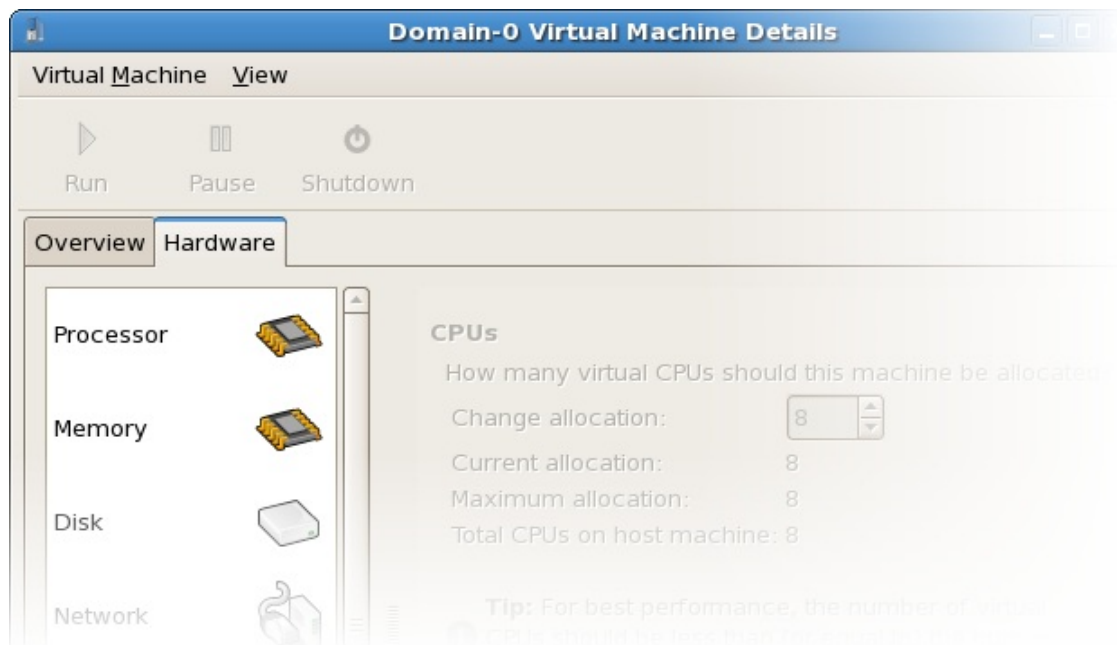


图 17.22. 显示 Virtual Machine Details Hardware

4. 在 **Hardware** 页，点击 **Processor** 可以查看或修改当前的处理器分配情况。



图 17.23. 显示处理器分配情况

5. 在 **Hardware** 窗口，点击 **Memory** 可以查看和修改当前的内存分配。

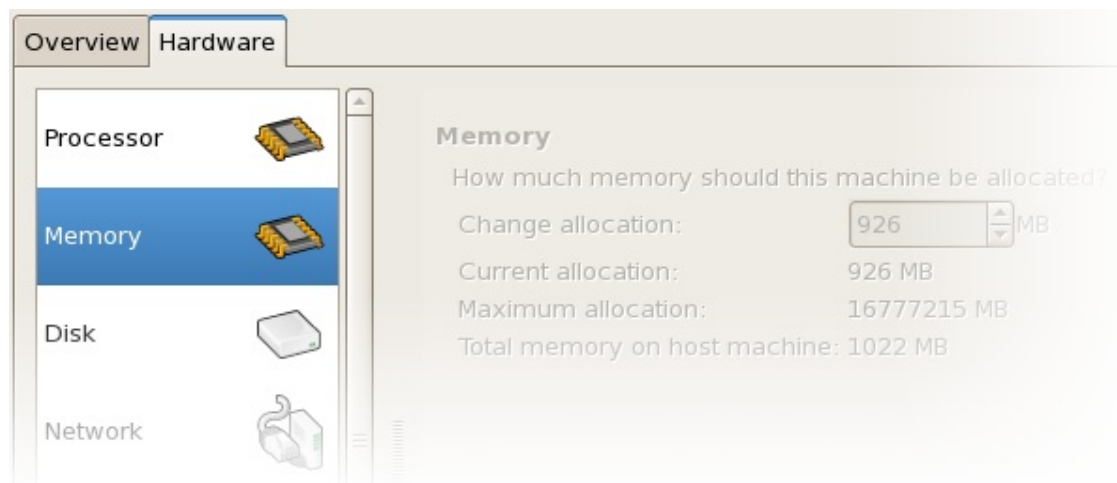


图 17.24. 显示内存分配情况

6. 在 **Hardware** 页，点击 **Disk** 可以查看或修改当前的硬盘配置。

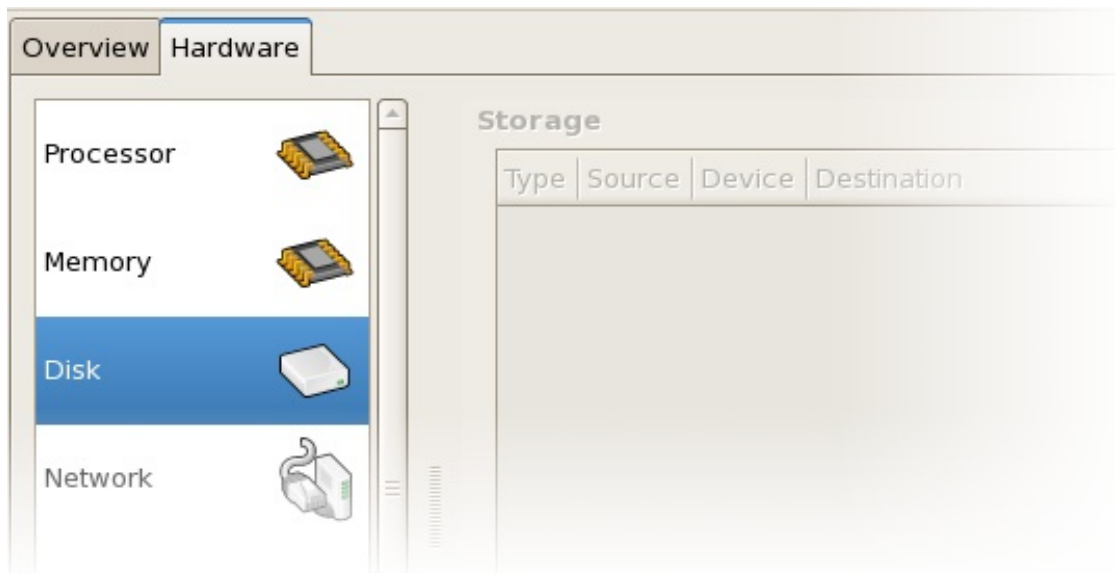


图 17.25. 显示磁盘配置

7. 在 **Hardware** 页，点击 **Network** 可以查看或修改当前的网络配置。

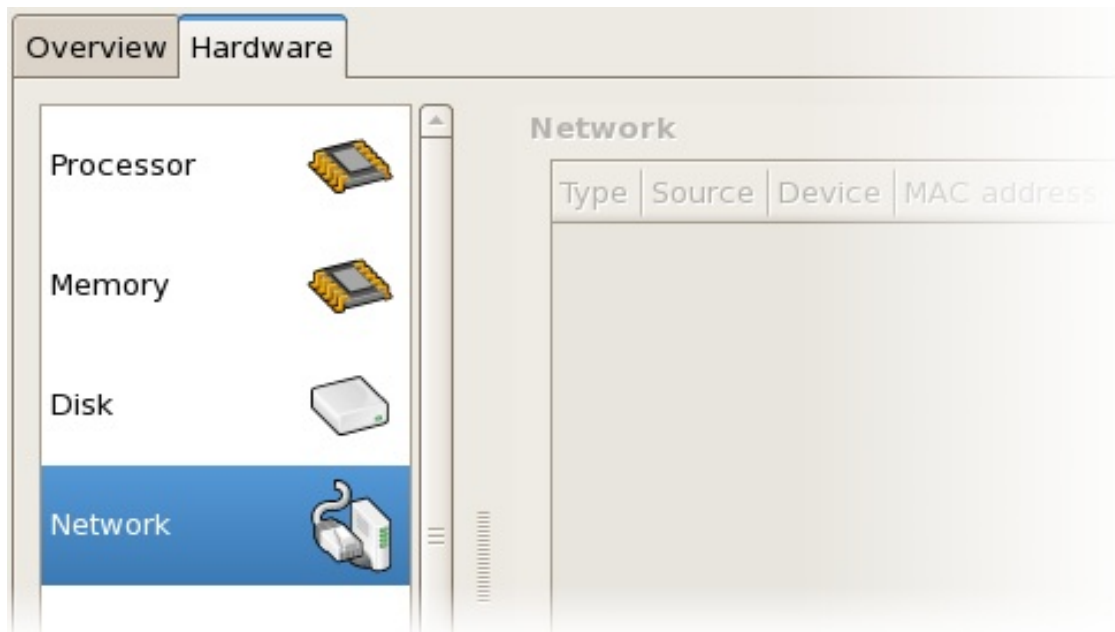


图 17.26. 显示网络配置

17.10. CONFIGURING STATUS MONITORING

你可以使用虚拟机管理者来修改虚拟系统的状态监控。

要配置状态监控并启用控制台：

1. 在 **Edit** 菜单，选择 **Preferences**。

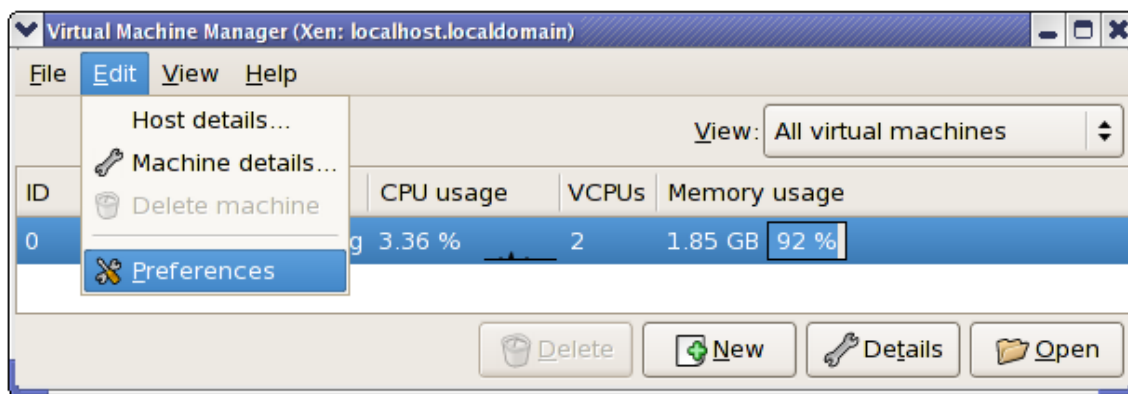


图 17.27. 改变虚拟机的首选项

虚拟机管理者首选项窗口将出现。

2. 在状态监控区选择框，指定你希望系统更新的时间间隔（以秒为单位）。

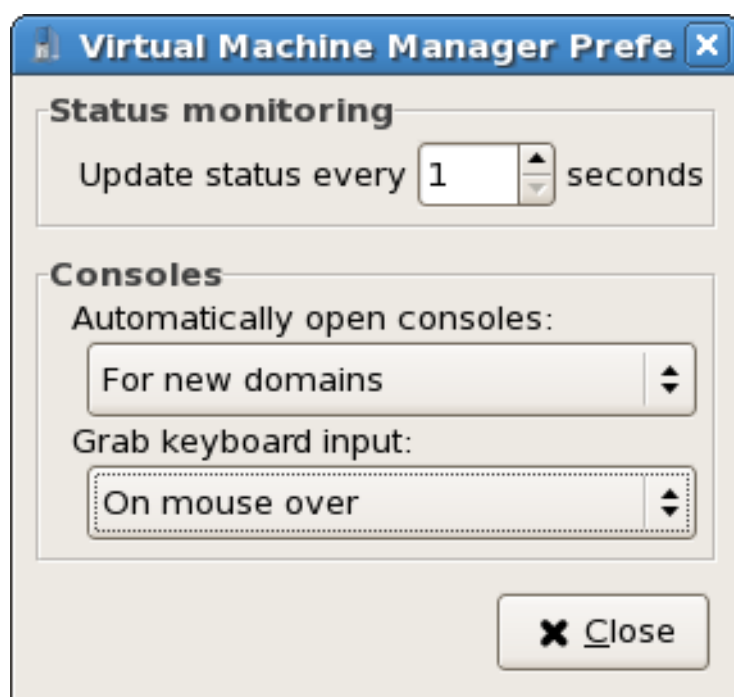


图 17.28. Configuring Status Monitoring

3. 在控制台区，指定怎样打开控制台和输入设备。

17.11. 显示 DOMAIN ID

查看系统里所有虚拟机的 domain ID：

1. 在 **View** 菜单，选定 **Domain ID** 复选框。

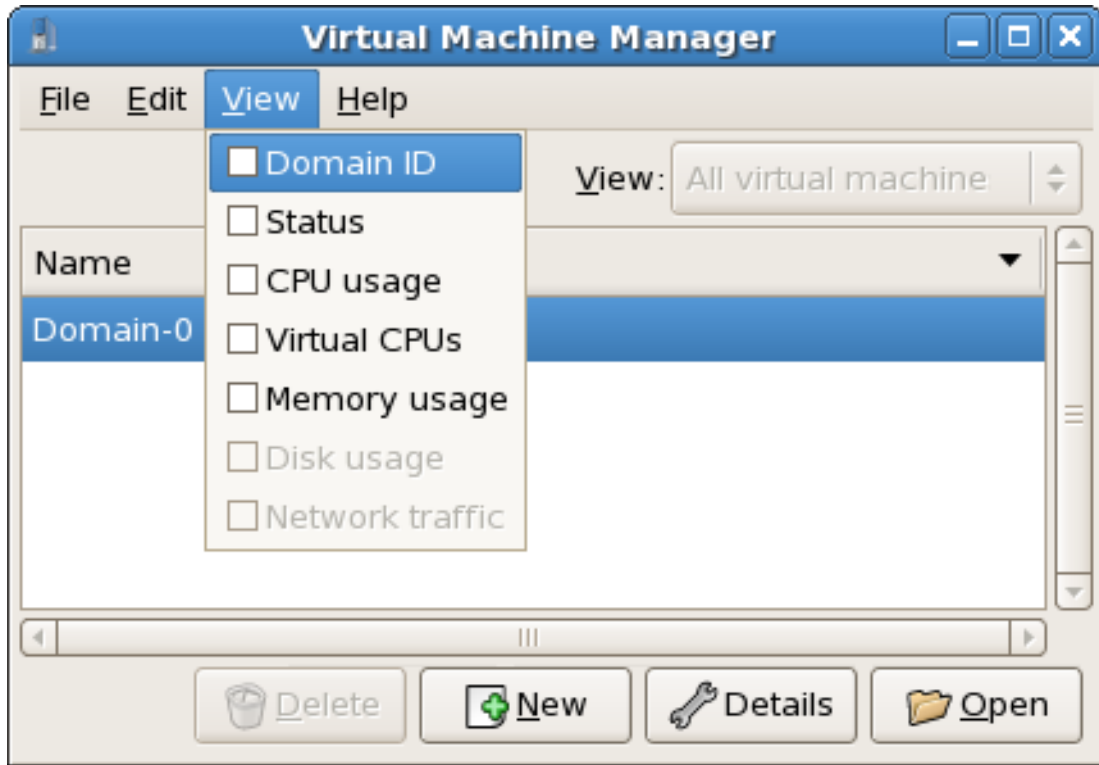


图 17.29. 显示 Domain-ID

2. The Virtual Machine Manager lists the Domain ID's for all domains on your system.

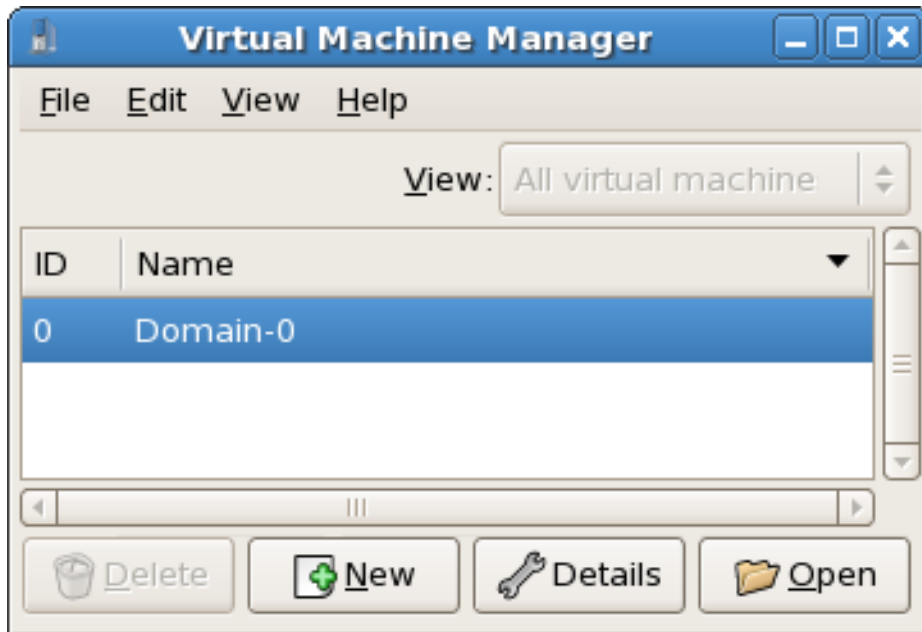


图 17.30. 显示 Domain-ID

17.12. 显示虚拟机状态

查看系统里所有虚拟机的状态：

1. 在 **View** 菜单，选择 **Status** 复选框。

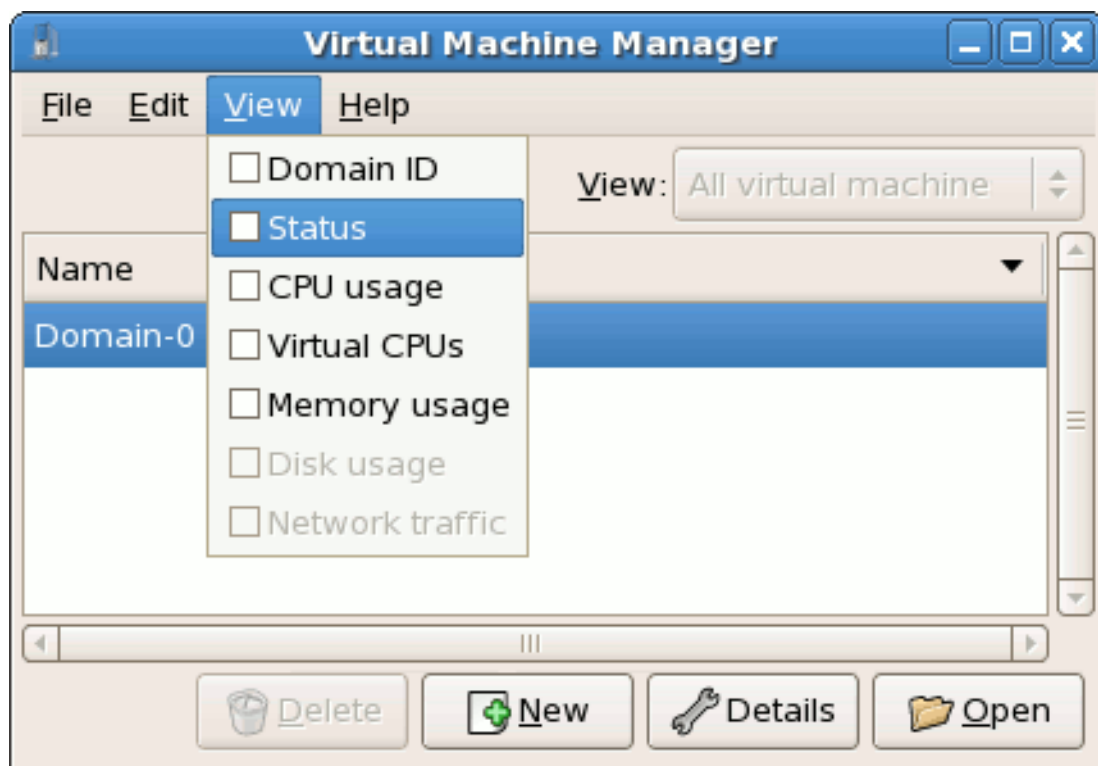


图 17.31. 显示虚拟机状态

- 虚拟机管理者列出了系统里所有虚拟机的状态。

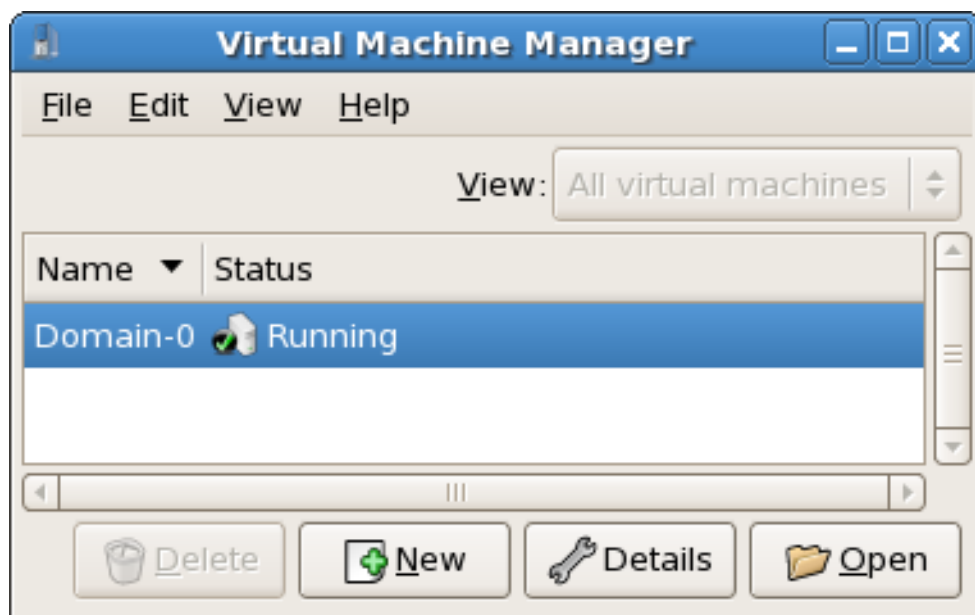


图 17.32. 显示虚拟机状态

17.13. 显示虚拟 CPU

查看系统里所有虚拟机的虚拟 CPU 的数量：

- 在 **View** 菜单，选定 **Virtual CPUs** 复选框。

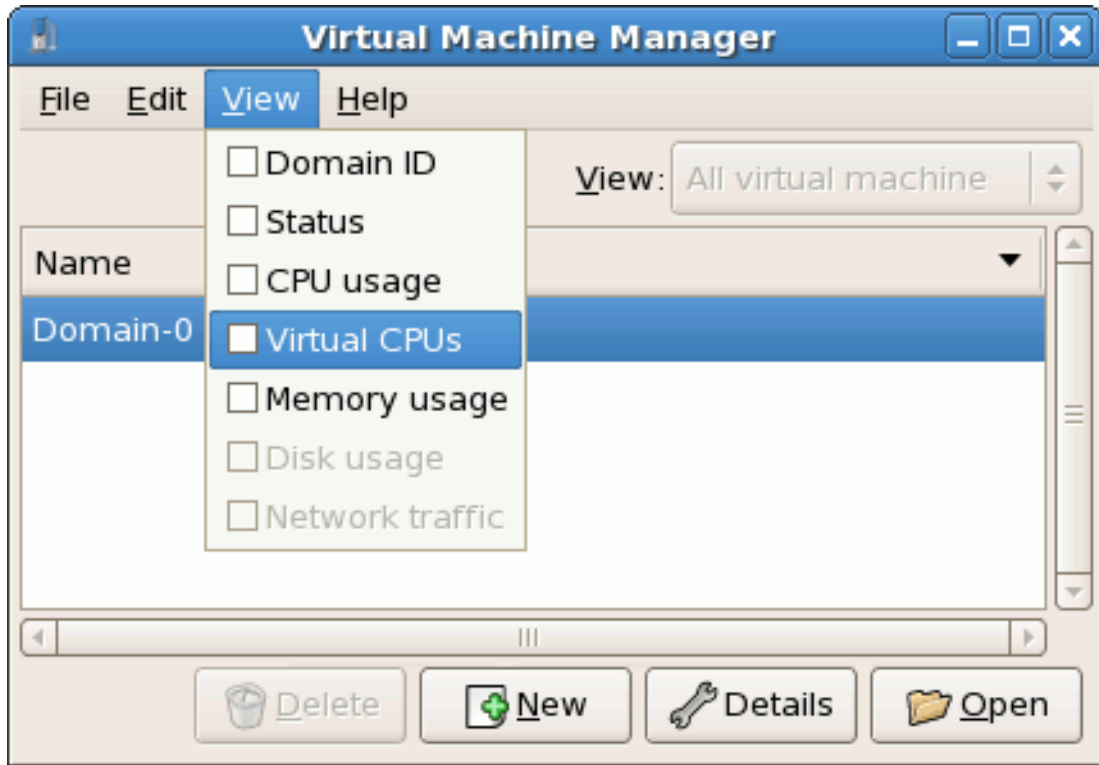


图 17.33. 显示虚拟 CPU

- 虚拟机管理者将列出系统里所有虚拟机的虚拟 CPU。

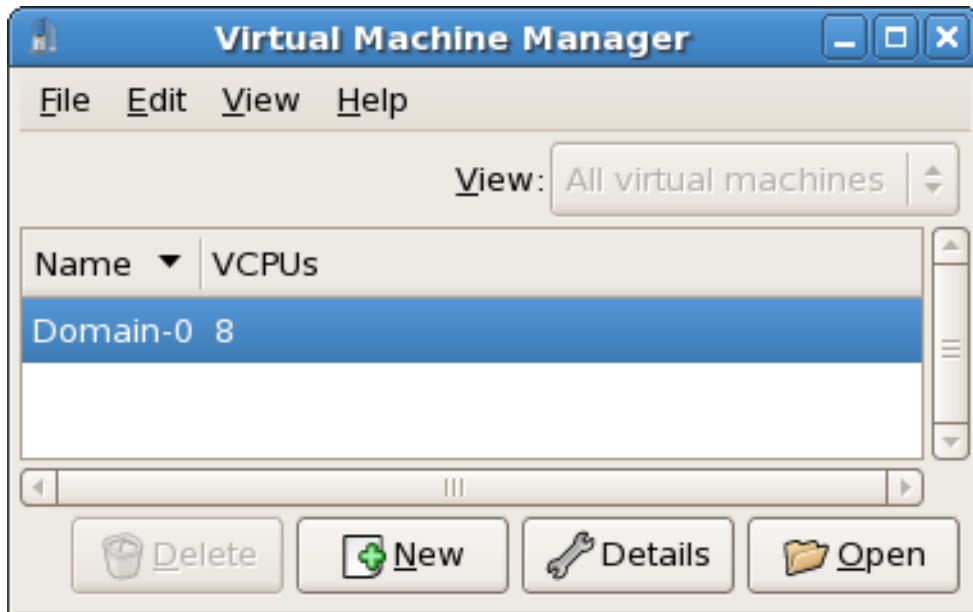


图 17.34. 显示虚拟 CPU

17.14. 显示 CPU 的使用情况

要查看系统里所有虚拟机的 CPU 使用情况：

- 在 **View** 菜单里，选定 **CPU Usage** 复选框。

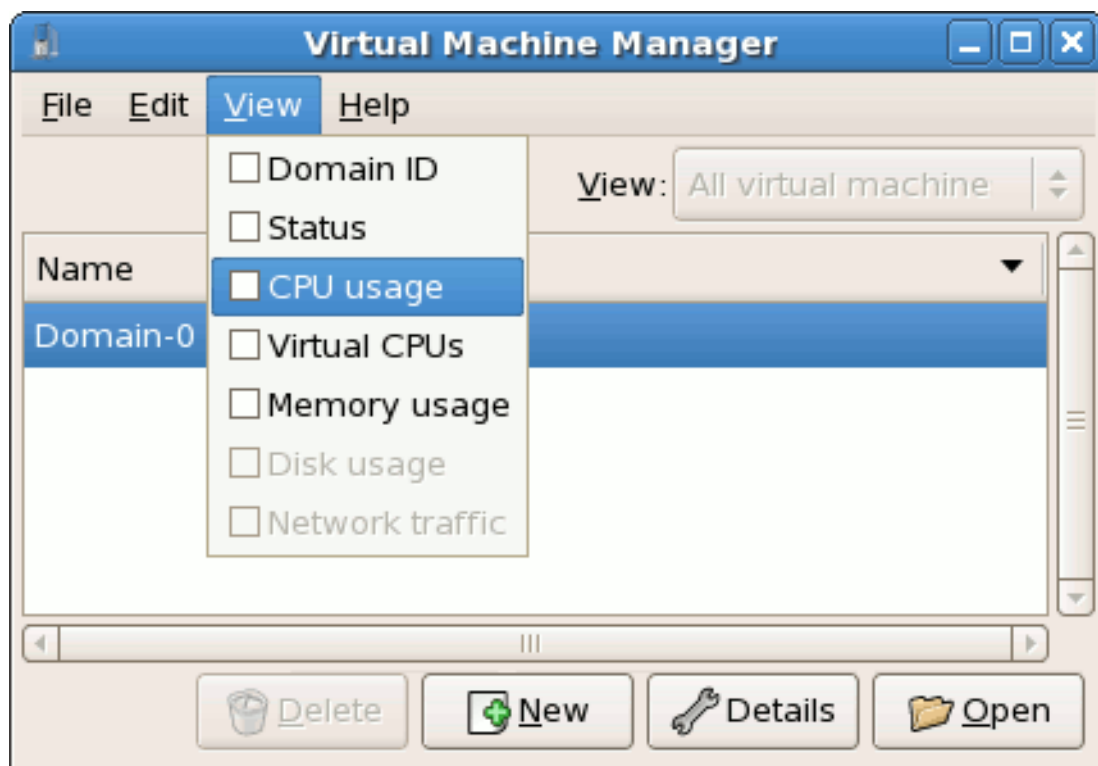


图 17.35. 显示 CPU 的使用情况

- 虚拟机管理者将列出系统里所有虚拟机的 CPU 使用百分比。

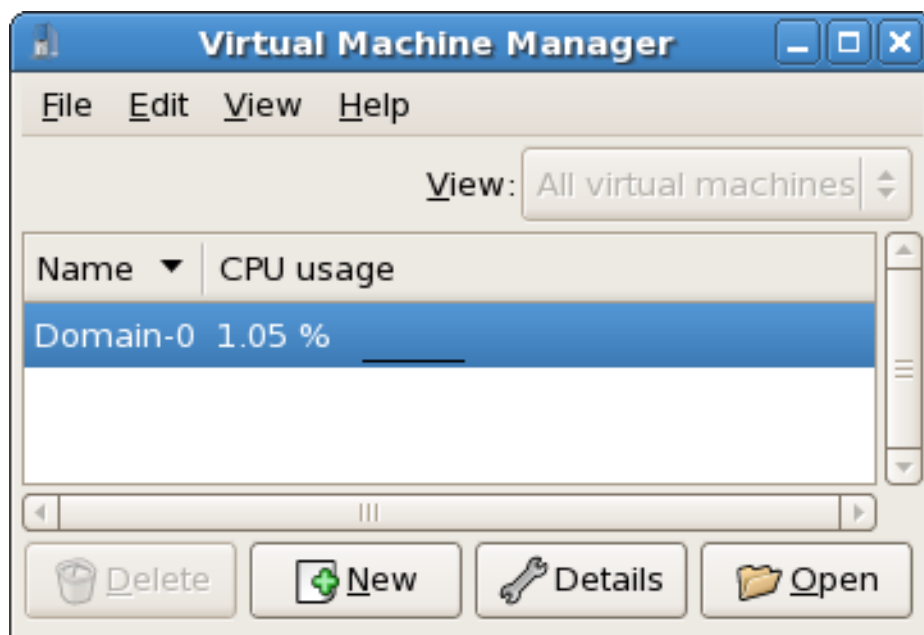


图 17.36. 显示 CPU 的使用情况

17.15. 显示内存使用情况

要查看系统里所有虚拟机的内存使用情况：

- 在 **View** 菜单里，选定 **Memory Usage** 复选框。

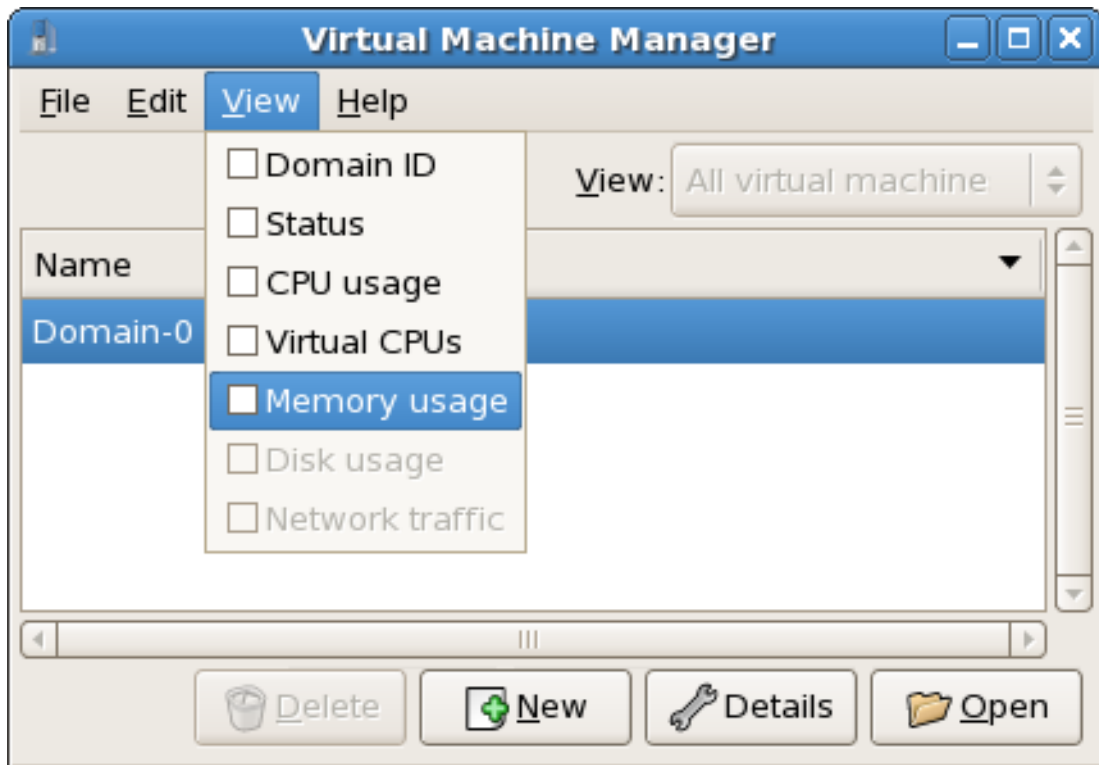


图 17.37. 显示内存使用情况

- 虚拟机管理者将列出系统里所有虚拟机的内存（以 MB 为单位）使用百分比。

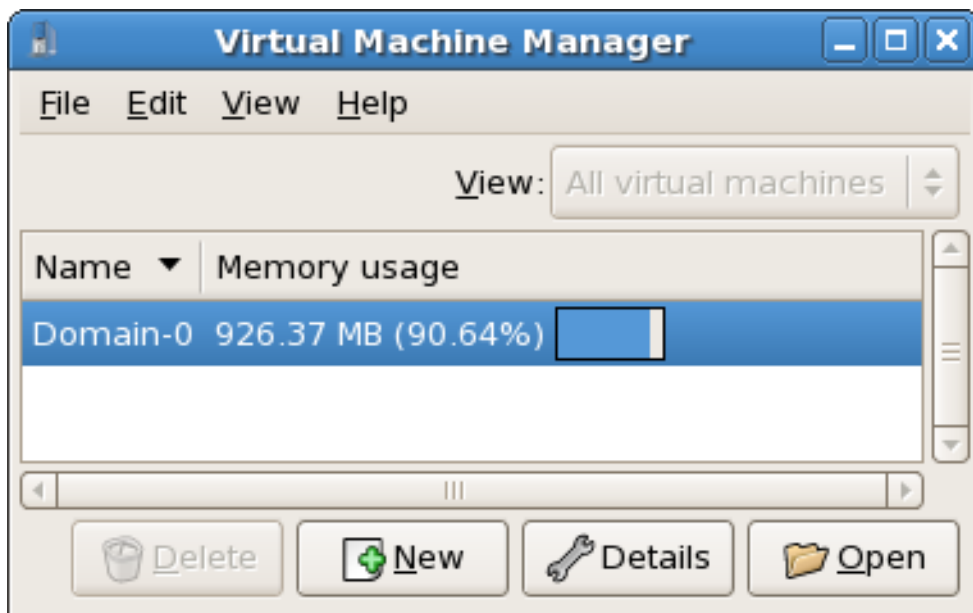


图 17.38. 显示内存使用情况

17.16. MANAGING A VIRTUAL NETWORK

To configure a virtual network on your system:

- From the **Edit** menu, select **Host Details**.

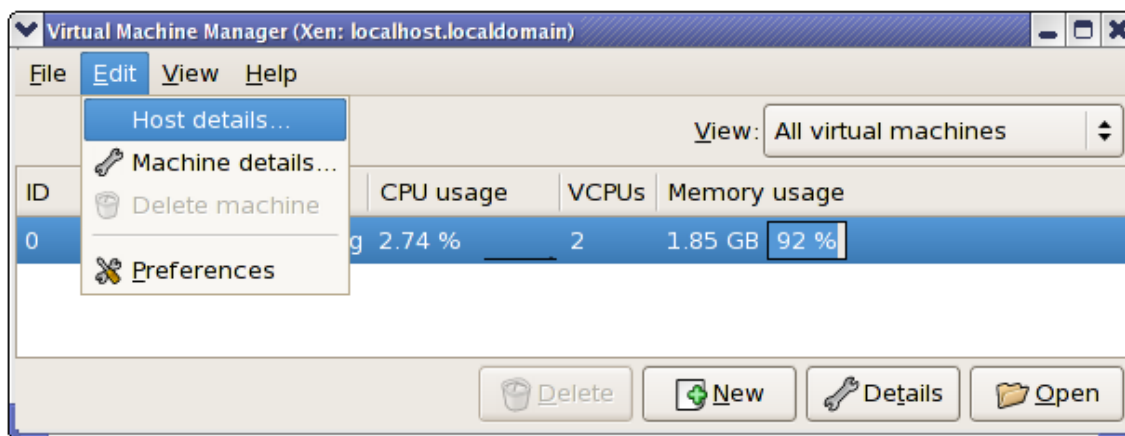


图 17.39. Selecting Host Details

2. This will open the Host Details menu. Click the **Virtual Networks** tab.

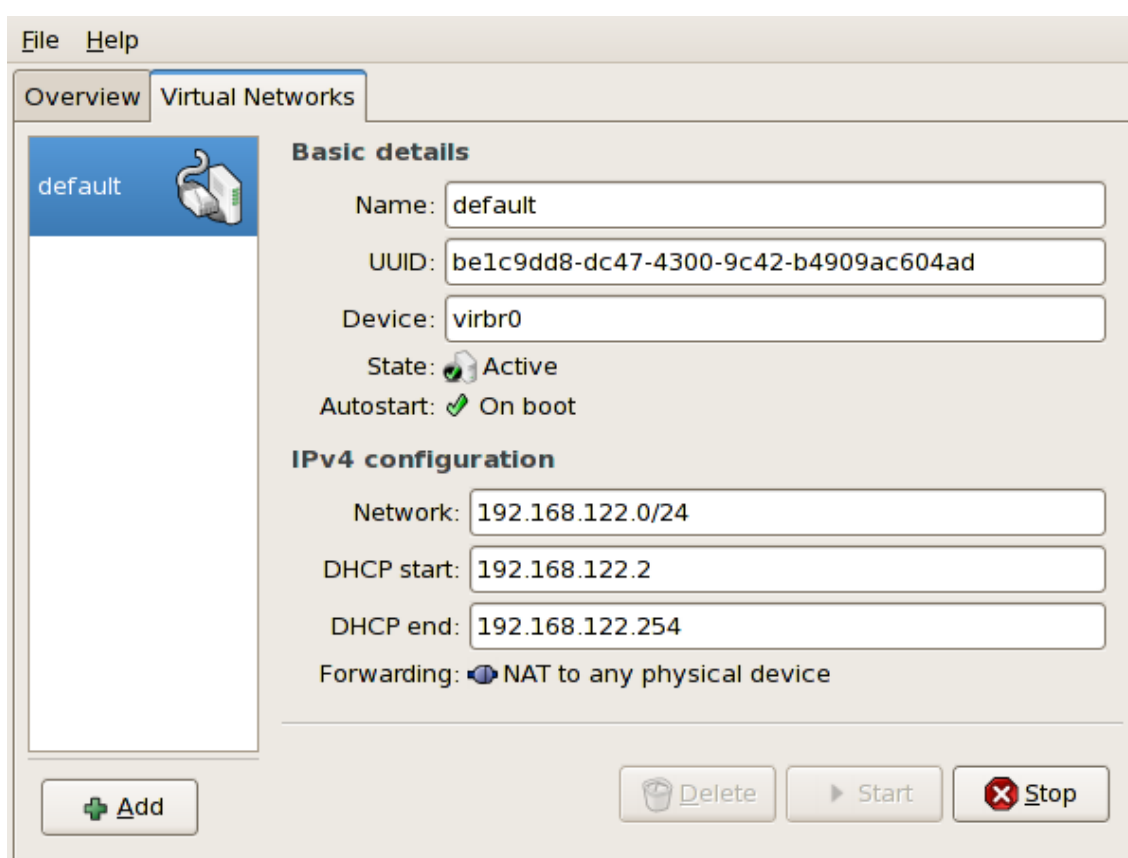


图 17.40. Virtual Network Configuration

3. All available virtual networks are listed on the left-hand box of the menu. You can edit the configuration of a virtual network by selecting it from this box and editing as you see fit.

17.17. CREATING A VIRTUAL NETWORK

To create a virtual network on your system:

1. Open the **Host Details** menu (refer to [第 17.16 节 “Managing a Virtual Network”](#)) and click the **Add** button.

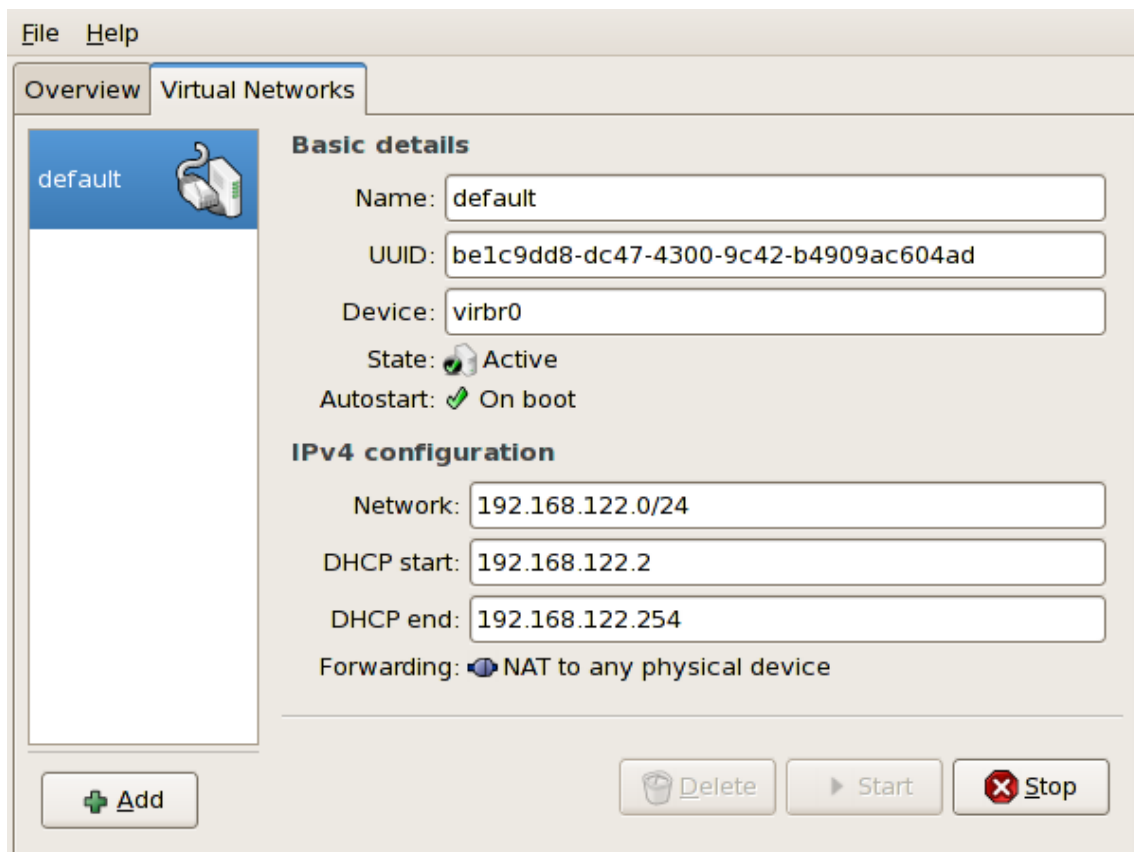


图 17.41. Virtual Network Configuration

This will open the **Create a new virtual network** menu. Click **Forward** to continue.

Creating a new virtual network

This assistant will guide you through creating a new virtual network. You will be asked for some information about the virtual network you'd like to create, such as:

- A **name** for your new virtual network
- The IPv4 **address** and **netmask** to assign
- The **address range** from which the **DHCP** server will allocate addresses for virtual machines
- Whether to **forward** traffic to the physical network



图 17.42. Creating a new virtual network

2. Enter an appropriate name for your virtual network and click **Forward**.

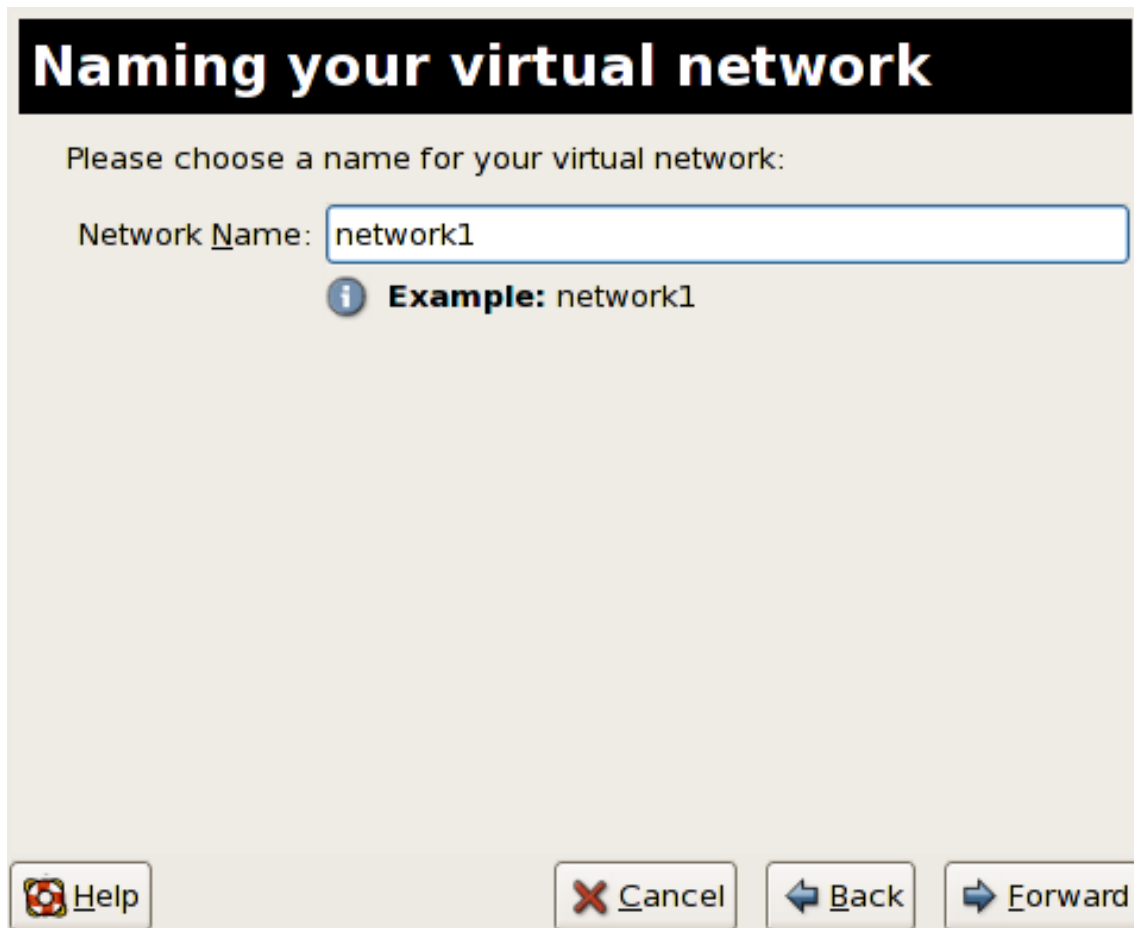


图 17.43. Naming your virtual network

3. Enter an IPv4 address space for your virtual network and click **Forward**.

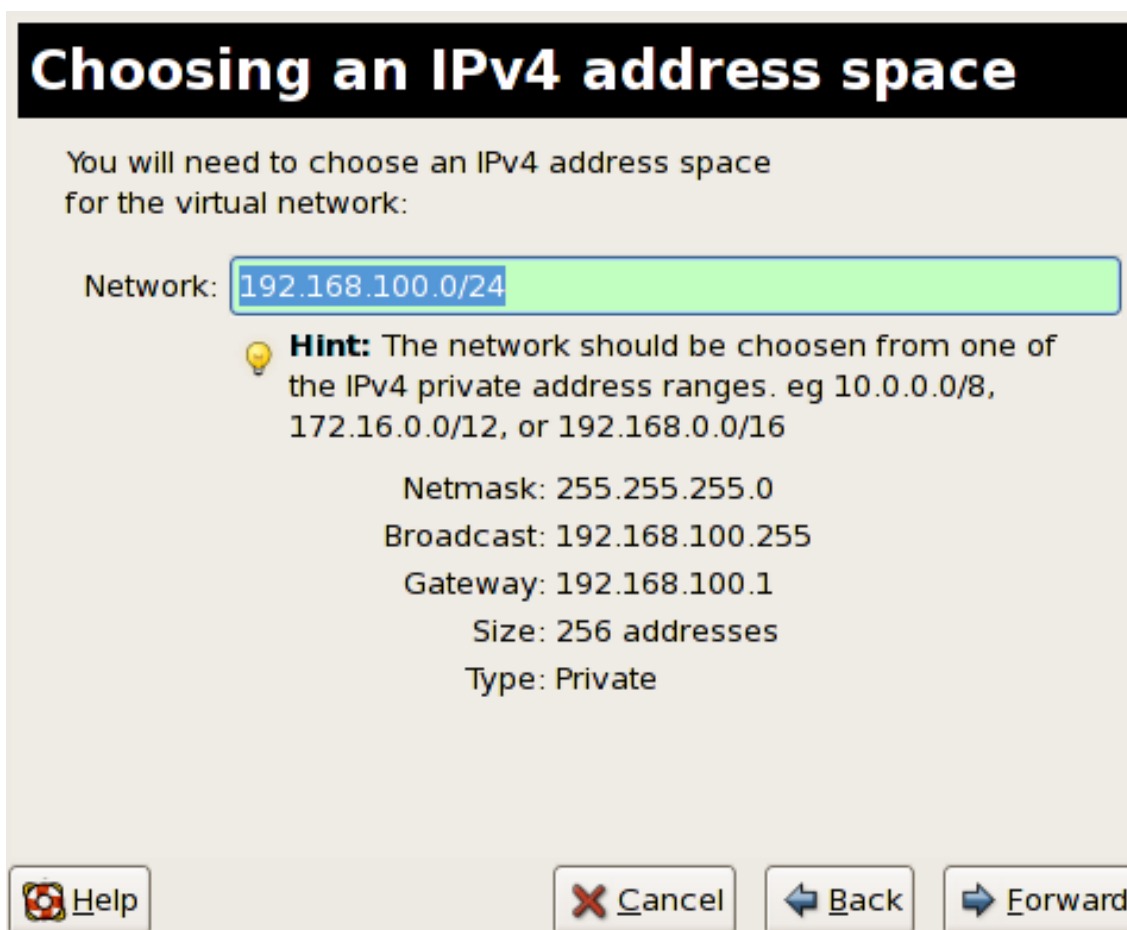


图 17.44. Choosing an IPv4 address space


4. Define the DHCP range for your virtual network by specifying a Start and End range of IP addresses. Click Forward to continue.

Selecting the DHCP range

Please choose the range of addresses the DHCP server can use to allocate to guests attached to the virtual network

Start:

End:

 **Tip:** Unless you wish to reserve some addresses to allow static network configuration in virtual machines, these parameters can be left with their default values.





 Help  Cancel  Back  Forward

图 17.45. Selecting the DHCP range

5. Select how the virtual network should connect to the physical network.

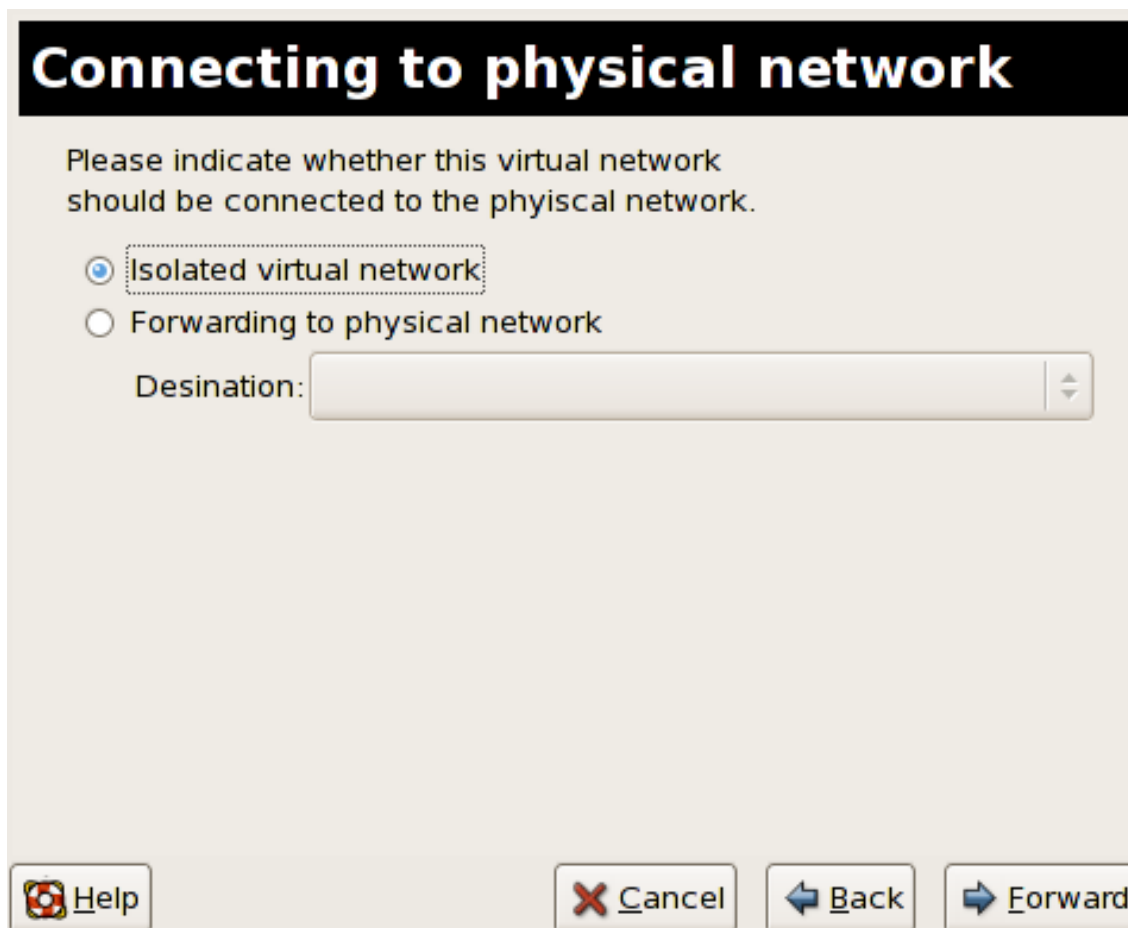


图 17.46. Connecting to physical network

If you select **Forwarding to physical network**, choose whether the **Destination** should be **NAT to any physical device** or **NAT to physical device eth0**.

Click **Forward** to continue.

6. You are now ready to create the network. Check the configuration of your network and click **Finish**.

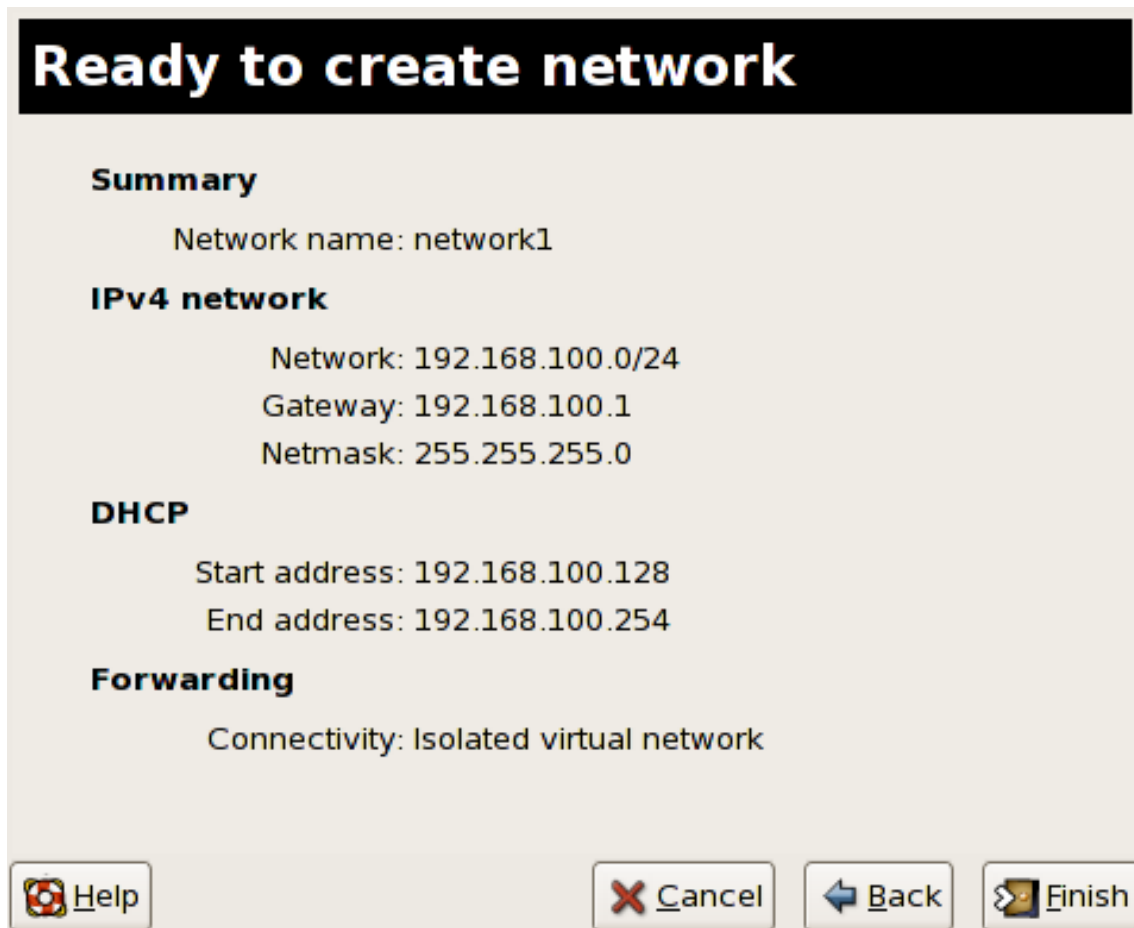


图 17.47. Ready to create network

7. The new virtual network is now available in the **Virtual Network** tab of the **Host Details** menu.

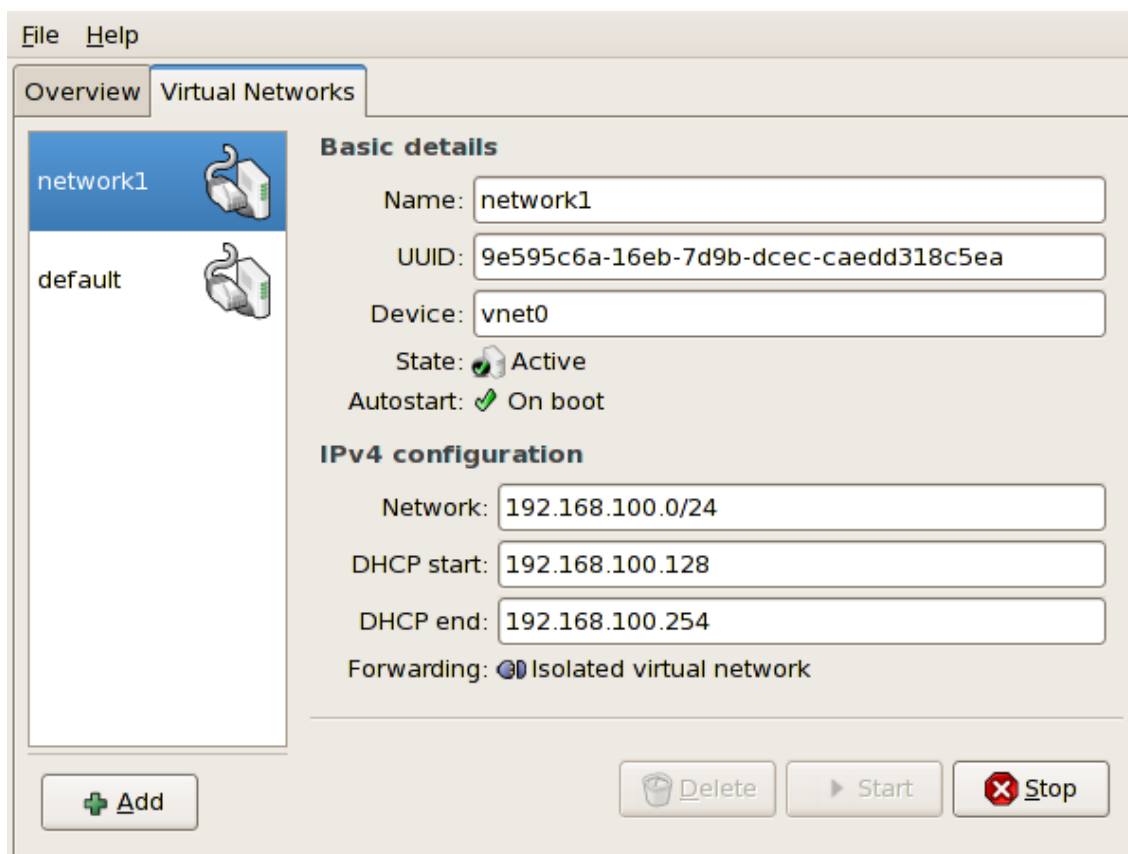


图 17.48. New virtual network is now available

第 18 章 红帽虚拟化系统的故障解除

本部分内容涵盖了你在安装、管理和日常操作过程中可能会遇到的潜在问题。它包括错误信息、日志文件的位置、系统工具和研究数据及分析问题的一般方法。

18.1. 日志文件概述和位置

When deploying Red Hat Enterprise Linux 5.1 with Virtualization into your network infrastructure, the host's Virtualization software uses many specific directories for important configuration, log files, and other utilities. All the Red Hat Virtualization logs files are standard ASCII files, and easily accessible with any ASCII based editor:

- 红帽虚拟化系统的主要配置目录是 `/etc/xen/`。这个目录包含了 `xend` 守护进程和其他的虚拟机配置文件。网络脚本文件也存放在此处（在 `/scripts` 目录）。
- 所有你用于故障解除目的的日志文件都存放在 `/var/log/xen` 目录。
- 你也应该知道所有虚拟机的基于文件的磁盘映像都缺省存放在 `/var/lib/xen` 目录里。
- 红帽虚拟化系统的 `/proc` 文件系统信息位于 `/proc/xen/` 目录。

18.2. 日志文件的描述

红帽虚拟化系统以 `xend` 守护进程和 `qemu-dm` 进程为特征，这两个工具都把多个日志文件写入到 `/var/log/xen/` 目录：

- `xend.log` 是包含 `xend` 守护进程收集的所有数据的日志文件，不管这数据是普通的系统事件，还是操作者执行的动作。所有虚拟机的操作（如创建、关闭、销毁等等）都在此出现。`xend.log` 通常是你跟踪事件或性能问题的第一个着手点。它包含错误信息的详细条目和条件。
- `xend-debug.log` 是包含 `xend` 和虚拟化子系统（如 `framebuffer`、`Python` 脚本等等）的事件错误记录的日志文件。
- `xen-hotplug-log` 是包含热插拔事件的数据的日志文件。如果设备或网络脚本没有被启动，事件将记录在这里。
- `qemu-dm.[PID].log` 是 `qemu-dm` 进程为每个完全虚拟化客户机创建的日志文件。当使用日志文件时，你必须用 `ps` 命令并挑选出 `qemu-dm` 的信息来获取 `qemu-dm` 的进程号。注意你必须用实际的 `qemu-dm` 进程的 PID 来代替 [PID]。

如果在使用虚拟机管理者时你遇到任何错误，你可以在 `/.virt-manager` 目录下的 `virt-manager.log` 文件里寻找生成的数据。注意每次你启动虚拟机管理者，它都会覆盖现有的日志文件内容。当出现系统错误后，在重启虚拟机管理者前，请确保备份好 `virt-manager.log` 文件。

18.3. 重要的目录位置

当在红帽虚拟化环境里跟踪错误和解决问题时，你应该记住其他几个的工具和日志文件：

- 虚拟机映像位于 `/var/lib/xen/images` 目录里。
- 当你重启 `xend` 守护进程时，它会更新 `/var/lib/xen/xend-db` 目录下的 `xend-database`。
- 虚拟机的转储文件（使用 `xm dump-core` 命令生成的）位于 `/var/lib/xen/dumps` 目录

下。

- `/etc/xen` 目录包含你用来管理系统资源的配置文件。`xend` 守护进程的配置文件是 `xend-config.sxp`，你可以使用这个文件来实施系统范围的修改并配置网络 callout。
- `proc` 命令是另一个允许你收集系统信息的资源。这些 `proc` 条目位于 `/proc/xen` 目录。

`/proc/xen/capabilities`

`/proc/xen/balloon`

`/proc/xen/xenbus/`

18.4. 故障解除工具

本部分总结了系统管理员程序、网络工具和高级调试工具（关于如何使用这些工具来配置红帽虚拟化服务的更多信息，请参考相关的配置文档）。你可以使用这些标准的系统管理员工具和日志来协助故障解除：

- `xentop`
- `xm dmesg`
- `xm log`
- `vmstat`
- `iostat`
- `lsof`

你可以使用这些标准的系统管理员工具和日志来协助故障解除：

- `Xen0profile`
- `systemTap`
- `crash`
- `sysrq`
- `sysrq t`
- `sysrq w`

你可以使用这些网络工具来协助故障解除：

- `ifconfig`
- `tcpdump`
- `brctl`

`brctl` 是一个在虚拟化 Linux 内核里检查和配置以太网桥配置的网络工具。在执行这些示例命令时，你必须拥有根用户权限：

```
# brctl show

bridge-name      bridge-id          STP  enabled  interfaces
-----
-----
xenbr0           8000.fefffffff    no   vif13.0
xenbr1           8000.ffffeffff    yes  pddummy0
xenbr2           8000.fffffffef    no   vif0.0

# brctl showmacs xenbr0

port-no          mac-addr           local?  ageing timer
-----
-----
1                fe:ff:ff:ff:ff:   yes     0.00
2                fe:ff:ff:fe:ff:   yes     0.00

# brctl showstp xenbr0

xenbr0

bridge-id        8000.fefffffff
designated-root   8000.fefffffff

root-port        0                  path-cost          0
max-age          20.00             bridge-max-age     20.00
hello-time       2.00              bridge-hello-time  2.00
forward-delay    0.00              bridge-forward-delay 0.00
ageing-time      300.01

hello-timer      1.43              tcn-timer          0.00
topology-change-timer 0.00             gc-timer           0.02
```

18.5. 利用日志进行故障解除

When encountering issues with installing Red Hat Virtualization, you can refer to the host system's two logs to assist with troubleshooting. The `xend.log` file contains the same basic information as when you run the `xm log` command. It resides in the `/var/log/` directory. Here is an example log entry for when you create a domain running a kernel:

```
[2006-12-27 02:23:02 xend] ERROR (SrvBase: 163) op=create: Error creating
domain: (0, 'Error')
Traceback (most recent call list)
File "/usr/lib/python2.4/site-packages/xen/xend/server/SrvBase.py" line 107
in_perform val = op_method (op,req)
File
"/usr/lib/python2.4/site-packages/xen/xend/server/SrvDomainDir.py line 71
in op_create
raise XendError ("Error creating domain: " + str(ex))
```

```
XendError: Error creating domain: (0, 'Error')
```

其他的日志文件，**xend-debug.log**，对于系统管理员也是非常有用的，因为它包含了比 **xend.log** 更为详细的信息。下面是同一个内核域创建问题里错误数据：

```
ERROR: Will only load images built for Xen v3.0
ERROR: Actually saw: GUEST_OS=netbsd, GUEST_VER=2.0, XEN_VER=2.0;
LOADER=generic, BSD_SYMTAB'
ERROR: Error constructing guest OS
```

当呼叫客户支持并和技术支持人员联系时，请记得附上这些日志文件的拷贝。

18.6. 用串行控制台进行故障解除

串行控制台在解决有难度的问题时很有用。如果虚拟化内核崩溃了，且监控程序（hypervisor）产生了一个错误，你没有办法在本机上来跟踪这个错误。然而，串行控制台允许你在远程主机上捕获它。你必须配置 Xen 主机把数据输出到串行控制台。然后你必须配置远程主机来捕获这些数据。要实现这种方法，你必须修改 **grub.conf** 文件里的这些选项来在 **com1 /dev/ttyS0** 上启用一个速率为 **38400-bps** 的串行控制台：

```
title Red Hat Enterprise Linux (2.6.18-8.2080_RHEL5xen0)
  root (hd0,2)
  kernel /xen.gz-2.6.18-8.el5 com1=38400,8n1
  module /vmlinuz-2.6.18-8.el5xen ro root=LABEL=/rhgb quiet console=xvc
console=tty xencons=xvc
  module /initrd-2.6.18-8.el5xen.img
```

The **sync_console** can help determine a problem that causes hangs with asynchronous hypervisor console output, and the "**pnpcapi=off**" works around a problem that breaks input on the serial console. The parameters "**console=ttyS0**" and "**console=tty**" means that kernel errors get logged with on both the normal VGA console and on the serial console. Then you can install and set up **ttwatch** to capture the data on a remote host connected by a standard null-modem cable. For example, on the remote host you could type:

```
ttwatch --name myhost --port /dev/ttyS0
```

这把 **/dev/ttyS0** 的输出导出至 **/var/log/ttwatch/myhost.log** 文件。

18.7. 对半虚拟化客户机控制台的访问

半虚拟化客户机操作系统自动地配置有一个虚拟文本控制台来把数据输入至 **Domain0** 操作系统，你可以在命令行键入下列命令：

```
xm console [domain name or number]
```

在这里 **domain100** 代表一个运行名或号码。你也可以使用虚拟机管理者来显示虚拟文本控制台。在 **Virtual Machine Details** 窗口中的 **View** 菜单里，选择 **Serial Console**。

18.8. 对完全虚拟化客户机控制台的访问

Full Virtualized guest operating systems automatically has a text console configured for use, but the difference is the kernel guest is not configured. To enable the guest virtual serial console to work with

the Full Virtualized guest, you must modify the guest's `grub.conf` file, and include the `'console=ttyS0 console=tty0'` parameter. This ensures that the kernel messages are sent to the virtual serial console (and the normal graphical console). If you plan to use the virtual serial console in a full virtualized guest, you must edit the configuration file in the `/etc/xen/` directory. On the host domain, you can then access the text console by typing:

```
xm console
```

你也可以使用虚拟机管理者来显示串行控制台。在 `Virtual Machine Details` 窗口中的 `View` 菜单里，选择 `Serial Console`。

18.9. 实施 LUN 持久化

如果你的系统没有使用多重路径 (`multipath`)，你可以使用 `udev` 来实施 lun 持久化。在实施之前，请确保你获得了正确的 `UUID`。一旦你获得了这些 `UUID`，你可以通过编辑 `/etc` 目录里的 `scsi_id` 文件来配置 lun 持久化。用文本编辑器里打开这个文件后，你必须注释掉如下一行：

```
# options=-b
```

然后用这个参数来代替它：

```
# options=-g
```

这告诉 `udev` 来监控所有返回 `UUID` 的系统 `SCSI` 设备，要获取系统 `UUID`，键入：

```
# scsi_id -g -s /block/sdc
```

输出应该如下所示：

```
[root@devices] # scsi_id -g -s /block/sdc
*3600a0b80001327510000015427b625e*
```

这个长字符串就是 `UUID`。既然 `UUID` 是与设备名无关的，检查每个设备路径来确保 `UUID` 号码对于每个设备都是相同的。当你把新设备加入到系统里时，`UUID` 不会改变。一旦你已经检查了设备路径，你必须创建设备命名规则。要创建这些规则，你必须编辑 `/etc/udev/rules.d` 目录里的 `20-names.rules` 文件。设备创建的命名规则应该遵循下面的格式：

```
# KERNEL="sd*", BUS="scsi", PROGRAM="sbin/scsi_id", RESULT="UUID",
NAME="devicename"
```

用上面检索的 `UUID` 替换现有的 `UUID` 和设备名。所以，这个规则应该象下面这样：

```
KERNEL="sd*", BUS="scsi", PROGRAM="sbin/scsi_id",
RESULT="3600a0b80001327510000015427b625e
", NAME="mydevicename"
```

这导致系统启用所有匹配 `/dev/sd*` 的设备来检查给定的 `UUID`。当它发现有匹配的设备时，它创建一个叫 `/dev/devicename` 的设备节点。在这个例子里，设备节点是 `/dev/mydevice`。最后，你需要把 `/etc` 目录里的 `rc.local` 文件附加在这个路径后：

```
/sbin/start_udev
```

用多重路径 (MULTIPATH) 实施 LUN 持久化

要在多重路径的环境里实施 lun 持久化，你必须定义多重路径设备的别名。例如，你必须编辑 `/etc/` 目录里的 `multipath.conf` 文件来定义四个设备别名：

```

multipath {
    wwid      3600a0b80001327510000015427b625e
    alias     oramp1
}
multipath {
    wwid      3600a0b80001327510000015427b6
    alias     oramp2
}
multipath {
    wwid      3600a0b80001327510000015427b625e
    alias     oramp3
}
multipath {
    wwid      3600a0b80001327510000015427b625e
    alias     oramp4
}

```

这就定义了四个 lun：`/dev/mpath/oramp1`、`/dev/mpath/oramp2`、`/dev/mpath/oramp3` 和 `dev/mpath/oramp4`。这些设备将位于 `/dev/mpath` 目录。这些 lun 的名字在重启后也将保持，因为它们都是基于 LUN 的 `wwid` 创建的别名。

18.10. SELINUX 的相关事宜

本部分内容包含了在红帽虚拟化环境里实施 SELinux 必须考虑的事情。当你部署系统修订和增加设备时，你必须相应地更新 SELinux 策略。要为客户机配置 LVM 卷，你必须为不同的底层块设备和卷组修改 SELinux 上下文。

```

# semanage fcontext -a -t xen_image_t -f -b /dev/sda2
# restorecon /dev/sda2

```

布尔值参数 `xend_disable_trans` 使 `xend` 在重启后进入 `unconfined` 模式。禁止对单个进程的保护比在整个系统里应用要更好。我们建议你不要把目录重新标记为将在其他地方使用的 `xen_image_t`。

18.11. 访问客户机磁盘映像里的数据

你可以使用两个单独的程序来访问客户机磁盘映像里的数据。在使用这些工具之前，你必须关闭客户机。从客户机和 `dom0` 访问文件系统有可能损害你的系统。

你可以使用 `kpartx` 程序来处理分区磁盘或 LVM 卷组：

```

yum install kpartx
kpartx -av /dev/xen/guest1
add map guest1p1 : 0 208782 linear /dev/xen/guest1 63
add map guest1p2: 0 16563015 linear /dev/xen/guest1 208845

```

要访问另一个分区的 LVM 卷，你必须用 `vgscan` 对 LVM 进行重新扫描并用 `vgchange -ay` 命令激活那个分区上的卷组（缺省为 `VolGroup00`）：

```
# kpartx -a /dev/xen/guest1
#vgscan
Reading all physical volumes . This may take a while...
Found volume group "VolGroup00" using metadata type lvm2
# vgchange -ay VolGroup00
2 logical volume(s) in volume group VolGroup00 now active.
# lvs
LV VG Attr Lsize Origin Snap% Move Log Copy%
LogVol00 VolGroup00 -wi-a- 5.06G
LogVol01 VolGroup00 -wi-a- 800.00M
# mount /dev/VolGroup00/LogVol00 /mnt/
....
#umount /mnt/
#vgchange -an VolGroup00
#kpartx -d /dev/xen/guest1
```

你必须记住用 **vgchange -an** 来使逻辑卷无效、用 **kpartx-d** 来删除分区且在完成后用 **losetup -d** 删除回路设备。

18.12. 常见的故障解除

当你试图启动 **xend** 服务时，什么也没有发生。键入 **xm list1** 后可以看到下面的输出：

```
Error: Error connecting to xend: Connection refused. Is xend running?
```

当你尝试手工运行 **xend start** 时，发现更多的错误：

```
Error: Could not obtain handle on privileged command interfaces (2 = No
such file or directory)
Traceback (most recent call last:)

File "/usr/sbin/xend/", line 33 in ?

from xen.xend.server import SrvDaemon

File "/usr/lib/python2.4/site-packages/xen/xend/server/SrvDaemon.py" ,
line 26 in ?

from xen.xend import XendDomain

File "/usr//lib/python2.4/site-packages/xen/xend/XendDomain.py" , line 33,
in ?

from xen.xend import XendDomainInfo

File "/usr/lib/python2.4/site-packages/xen/xend/image.py" , line37, in ?

import images

File "/usr/lib/python2.4/site-packages/xen/xend/image.py" , line30, in ?

xc = xen.lowlevel.xc.xc ()

RuntimeError: (2, 'No such file or directory' )
```

最可能的原因就是你把主机启动到了非 **xen-hypervisor** 的内核。要解决这个问题，你必须在引导时选择 **xen-hypervisor** 内核（或在 **grub.conf** 文件里把 **xen-hypervisor** 内核设置为缺省选项）。

18.13. 回路设备错误

如果你使用基于文件的客户机映像，你可以增加回路设备的数量（缺省是允许 8 个活动的回路设备）。如果你需要多于 8 个的基于文件的客户机回路设备，你必须修改 **/etc/modprobe.conf** 文件。当修改 **modprobe.conf** 文件时，你必须包含下面这行：

```
options loop max_loop=64
```

你可以用对应你的配置的其他数字来代替 **64**。注意，在系统里使用回路设备支持的客户机可能不是理想的选择。相反，你可以把 **phy: block device** 或 **tap:aio** 设备用于半虚拟化系统，把 **phy: device** 或 **file: file** 用于完全虚拟化系统。

18.14. 客户机创建错误

When you attempt to create a guest, you receive an "Invalid argument" error message. This usually means that the kernel image you are trying to boot is incompatible with the hypervisor. An example of this would be if you were attempting to run a non-PAE FC5 kernel on a PAE only FC6 hypervisor.

当你用 **yum update** 来更新内核时，**grub.conf** 里的缺省内核选项切换回普通内核而不是虚拟化内核。

要解决这个问题，你必须修改 **/etc/sysconfig/kernel/** 目录里的缺省内核 RPM。你必须确保在 **gb.conf** 文件里 **kernel-xen** 参数被设置为缺省选项。

18.15. 串行控制台错误

你在串行控制台里接受不到任何输出。要解决这个问题，你必须把 **grub.conf** 里的串口参数修改为：

```
serial --unit=1 --speed=115200

title RHEL5 i386 Xen (2.6.18-1.2910.el5xen)
root (hd0, 8)
kernel /boot/xen.gz-2.6.18-1.2910.el5 com2=115200,8n1
module /boot/vmlinuz-2.6.18-1.2910.el5xen to root=LABEL=RHEL5_i386
console=tty console=ttyS1115200
module /boot/initrd-2.8.6.18-12910.el5xen.img

title RHEL5 i386 xen (2.6.18.-1.2910.el5xen)
root (hd0, 8)
kernel /boot/xen.gz-2.6.18-1.2910.el5 com2=115200 console=com2l
module /boot/vmlinuz2.6.18-1.2910.el5xen to root=LABEL=RHEL5_i386
console=xvc xencons=xvc
module /boot/ititrd-2.6.18-1.2910.el5xen.img
```

这些对 **grub.conf** 的修改应该能使串行控制台正常工作。你应该能够使用任何数量的 **ttys**，如 **ttys0**。

18.16. 网桥错误

Red Hat Virtualization can configure multiple Virtualization network bridges to use with multiple ethernet cards. To successfully configure multiple network bridges for ethernet cards, you must configure the second network interface by either using the system-config-network TUI/GUI, or by creating a new configuration file in `/etc/sysconfig/network-scripts`. You should use a process to setup multiple Xen bridges. This is an example config file for a second NIC called 'eth1' :

```

#/etc/sysconfig/network-scripts/fcfg-eth1
DEVICE=eth1
BOOTPROTO=static
ONBOOT=yes
USERCTL=no
IPV6INIT=no
PEERDNS=yes
TYPE=Ethernet
NETMASK=255.255.255.0
IPADDR=10.1.1.1
GATEWAY=10.1.1.254
ARP=yes

```

把 `/etc/xen/scripts/network-bridge` 复制到 `/etc/xen/scripts/network-bridge.xen` 。

Edit `/etc/xen/xend-config.sxp` and add a line to your new network bridge script (this example uses "network-virtualization-multi-bridge").

在 `xend-config.sxp` 文件里，加入的新行应该与新脚本相关：

```
network-script network-xen-multi-bridge
```

请确保不要注释下面的一行：

```
network-script network-bridge
```

如果你想要创建多个 Xen 网桥，你必须创建一个自定义的脚本。下面的例子创建了两个 Xen 网桥（称为 `xenbr0` 和 `xenbr1`）并相应地把它们附加到 `eth1` 和 `eth0`：

```

#!/bin/sh
# network-xen-multi-bridge
# Exit if anything goes wrong
set -e
# First arg is operation.
OP=$1
shift
script=/etc/xen/scripts/network-bridge.xen
case ${OP} in
start)
$script start vifnum=1 bridge=xenbr1 netdev=eth1
$script start vifnum=0 bridge=xenbr0 netdev=eth0
;;
stop)
$script stop vifnum=1 bridge=xenbr1 netdev=eth1
$script stop vifnum=0 bridge=xenbr0 netdev=eth0
;;
status)

```



```

$script status vifnum=1 bridge=xenbr1 netdev=eth1
$script status vifnum=0 bridge=xenbr0 netdev=eth0
;;
*)
echo 'Unknown command: ' ${OP}
echo 'Valid commands are: start, stop, status'
exit 1
esac

```

如果你想创建其他的网桥，你可以使用示例脚本并复制/粘贴相应的文件。

18.17. 笔记本配置

The task of configuring your RHEL 5.1 loaded laptop for use on a network environment, presents a number of potential challenges. Most WiFi and wired connections switch constantly during any given day, and Red Hat Virtualization assumes it has access to the same interface consistently. This results in the system performing ifup/ifdown calls to the network interface in use by Red Hat Virtualization. WiFi cards are not the ideal network connection method since Red Hat Virtualization uses the default network interface.

The idea here is to create a 'dummy' network interface for Red Hat Virtualization to use.

This technique allows you to use a hidden IP address space for your guests and Virtual Machines. To do this operation successfully, you must use static IP addresses as DHCP does not listen for IP addresses on the dummy network. You also must configure NAT/IP masquerading to enable network access for your guests and Virtual Machines. You should attach a static IP when you create the 'dummy' network interface.

在这个例子里，接口被称为 `dummy0`，IP 地址为 10.1.1.1。脚本被称作 `ifcfg-dummy0` 并位于 `/etc/sysconfig/network-scripts/` 目录下。

```

DEVICE=dummy0
BOOTPROTO=none
ONBOOT=yes
USERCTL=no
IPV6INIT=no
PEERDNS=yes
TYPE=Ethernet
NETMASK=255.255.255.0
IPADDR=10.1.1.1
ARP=yes

```

你应该把 `xenbr0` 绑定至 `dummy0`，这样即使和物理网络断开，这也可以允许网络连接。

You will need to make additional modifications to the `xend-config.sxp` file. You must locate the (`network-script 'network-bridge' bridge=xenbr0`) section and add include this in the end of the line:

```
netdev=dummy0
```

You must also make some modifications to your guest's domU networking configuration to enable the default gateway to point to `dummy0`. You must edit the DomU 'network' file that resides in the `/etc/sysconfig/` directory to reflect the example below:

```
NETWORKING=yes
HOSTNAME=localhost.localdomain
GATEWAY=10.1.1.1

IPADDR=10.1.1.10
NETMASK=255.255.255.0
```

启用 `domain0` 里的 NAT 是个好主意，这样 `domU` 就可以访问公共网络。用这种方式，即使是无线用户也可以绕过红帽虚拟化系统的无线限制。如下例所示，要实现这种方法，你必须修改 `/etc/rc3.d` 目录里的 `S99XenLaptopNAT` 文件。

```
#!/bin/bash
#
# XenLaptopNAT Startup script for Xen on Laptops
#
# chkconfig: - 99 01
# description: Start NAT for Xen Laptops
#
# PATH=/usr/bin:/sbin:/bin:/usr/sbin
# export PATH
GATEWAYDEV=`ip route | grep default | awk {'print $5'}`
iptables -F
case "$1" in
start)
if test -z "$GATEWAYDEV"; then
echo "No gateway device found"
else
echo "Masquerading using $GATEWAYDEV"
/sbin/iptables -t nat -A POSTROUTING -o $GATEWAYDEV -j MASQUERADE
fi
echo "Enabling IP forwarding"
echo 1 > /proc/sys/net/ipv4/ip_forward
echo "IP forwarding set to `cat /proc/sys/net/ipv4/ip_forward`"
echo "done."
;;
*)
echo "Usage: $0 {start|restart|status}"
;;
esac
```

如果你想在引导时自动设置网络，你必须创建 `/etc/rc3.d/S99XenLaptopNAT` 的软链接。

在修改 `modprobe.conf` 文件时，你必须包括下面的行：

```
alias dummy0 dummy
options dummy numdummies=1
```

18.18. STARTING DOMAINS AUTOMATICALLY DURING SYSTEM BOOT

Starting Domains Automatically During System Boot

你可以配置客户机在在系统引导时自动启动域。要实现这种方式，你必须修改 `/etc/xen/auto` 里的符号链接。这些文件指向你需要自动启动的客户机配置文件。启动过程是按数字排列的，客户机的数字越大，引导过程所需的时间越长。下例告诉你怎样为客户机 `rhel5vm01` 使用符号链接：

```
[root@python xen]# cd /etc/xen
[root@python xen]# cd auto
[root@python auto]# ls
[root@python auto]# ln -s ../rhel5vm01 .
[root@python auto]# ls -l

lrwxrwxrwx 1 root root 14 Dec 14 10:02 rhel5vm01 -> ../rhel5vm01

[root@python auto]#
```

18.19. 修改 DOMAIN0

To use Red Hat Virtualization to manage domain0, you will constantly making changes to the `grub.conf` configuration file, that resides in the `/etc` directory. Because of the large number of domains to manage, many system administrators prefer to use the 'cut and paste' method when editing `grub.conf`. If you do this, make sure that you include all five lines in the Virtualization entry (or this will create system errors). If you require Xen hypervisor specific values, you must add them to the 'xen' line. This example represents a correct `grub.conf` Virtualization entry:

```
# boot=/dev/sda/
default=0
timeout=15
#splashimage=(hd0, 0)/grub/splash.xpm.gz
hiddenmenu
serial --unit=0 --speed=115200 --word=8 --parity=no --stop=1
terminal --timeout=10 serial console
title Red Hat Enterprise Linux Server (2.6.17-1.2519.4.21. el5xen)
    root (hd0, 0)
    kernel /xen.gz-2.6.17-1.2519.4.21.el5 com1=115200,8n1
    module /vmlinuz-2.6.17-1.2519.4.21el5xen ro root=/dev/VolGroup00/LogVol100
    module /initrd-2.6.17-1.2519.4.21.el5xen.img
```

For example, if you need to change your dom0 hypervisor's memory to 256MB at boot time, you must edit the 'xen' line and append it with the correct entry, '`dom0_mem=256M`'. This example represents the respective `grub.conf` xen entry:

```
# boot=/dev/sda
default=0
timeout=15
#splashimage=(hd0,0)/grubs/splash.xpm.gz
hiddenmenu
serial --unit=0 --speed =115200 --word=8 --parity=no --stop=1
terminal --timeout=10 serial console
title Red Hat Enterprise Linux Server (2.6.17-1.2519.4.21. el5xen)
    root (hd0,0)
    kernel /xen.gz-2.6.17-1.2519.4.21.el5 com1=115200,8n1 dom0_mem=256MB
    module /vmlinuz-2.6.17-1.2519.4.21.el5xen ro
    root=/dev/VolGroup00/LogVol100
    module /initrd-2.6.17-1.2519.4.21.el5xen.img
```

18.20. 客户机配置文件

When you install new guests using `virt-manager` (or `virt-install`) tool(s) from Red Hat Enterprise Linux 5.1 with Virtualization, the guests configuration files (located in the `/etc/xen` directory) get modified and setup automatically. This configuration file example is for a para-virtualized guest:

```
name = "rhel5vm01"
memory = "2048"
disk = ['tap:aio:/xen/images/rhel5vm01.dsk,xvda,w',]
vif = ["type=ieomu, mac=00:16:3e:09:f0:12 bridge=xenbr0',
      "type=ieomu, mac=00:16:3e:09:f0:13 ]
vnc = 1
vncunused = 1
uuid = "302bd9ce-4f60-fc67-9e40-7a77d9b4e1ed"
bootloader = "/usr/bin/pygrub"
vcpus=2
on_reboot = "restart"
on_crash = "restart"
```

Note that the `serial="pty"` is the default for the configuration file. This configuration file example is for a fully-virtualized guest:

```
name = "rhel5u5-86_64"
builder = "hvm"
memory = 500
disk = ['file:/xen/images/rhel5u5-x86_64.dsk.hda,w']
vif = [ 'type=ioemu, mac=00:16:3e:09:f0:12, bridge=xenbr0', 'type=ieomu,
mac=00:16:3e:09:f0:13, bridge=xenbr1']
uuid = "b10372f9-91d7-ao5f-12ff-372100c99af5"
device_model = "/usr/lib64/xen/bin/qemu-dm"
kernel = "/usr/lib/xen/boot/hvmloader/"
vnc = 1
vncunused = 1
apic = 1
acpi = 1
pae = 1
vcpus =1
serial ="pty" # enable serial console
on_boot = 'restart'
```

18.21. 克隆客户机配置文件

You can copy (or clone) an existing configuration file to create an all new guest. You must modify the name parameter of the guests' configuration file. The new, unique name then appears in the hypervisor and is viewable by the management utilities. You must generate an all new UUID as well (using the `uuidgen(1)` command). Then for the `vif` entries you must define a unique MAC address for each guest (if you are copying a guest configuration from an existing guest, you can create a script to handle it). For the xen bridge information, if you move an existing guest configuration file to a new host, you must update the `xenbr` entry to match your local networking configuration. For the Device entries, you must modify the entries in the '`disk=`' section to point to the correct guest image.

You must also modify these system configuration settings on your guest. You must modify the `HOSTNAME` entry of the `/etc/sysconfig/network` file to match the new guest's hostname.

你必须修改 `/etc/sysconfig/network-scripts/ifcfg-eth0` 文件的 `HWADDR` 地址，使之和 `ifconfig eth0` 的输出相符合。如果你想使用静态 IP 地址，你必须修改 `IPADDR` 条目。

18.22. 创建生成 MAC 地址的脚本

红帽虚拟化系统可以在创建时为每个虚拟机生成一个 MAC 地址。虽然在同一子网里几乎可以有无限个可用选择，你仍有可能获得相同的 MAC 地址。要绕过这个问题，你可以编写一个脚本来生成 MAC 地址。下面是一个生成 MAC 地址的包含参数的脚本示例：

```
#!/usr/bin/python
# macgen.py script generates a MAC address for Xen guests
#
import random
mac = [ 0x00, 0x16, 0x3e,
        random.randint(0x00, 0x7f),
        random.randint(0x00, 0xff),
        random.randint(0x00, 0xff) ]
print ':'.join(map(lambda x: "%02x" % x, mac))

Generates e.g.:
00:16:3e:66:f5:77
to stdout
```

18.23. 配置虚拟机的 LIVE 移植

Red Hat Virtualization can migrate virtual machines between other servers running Red Hat Enterprise Linux 5.1 with Virtualization. Further, migration is performed in an offline method (using the `xm migrate` command). Live migration can be done from the same command. However there are some additional modifications that you must do to the `xend-config` configuration file. This example identifies the entries that you must modify to ensure a successful migration:

(xend-relocation-server yes)

The default for this parameter is 'no', which keeps the relocation/migration server deactivated (unless on a trusted network) and the domain virtual memory is exchanged in raw form without encryption.

(xend-relocation-port 8002)

这个参数设置了 `xend` 用于移植的端口。请确保把它前面的注释去掉。

(xend-relocation-address)

在你启用了 `xend-relocation-server` 后，这个参数是侦听 relocation 套接字连接的地址。它可以把移植限制到特定的接口。

(xend-relocation-hosts-allow)

This parameter controls the host that communicates with the relocation port. If the value is empty, then all incoming connections are allowed. You must change this to a space-separated sequences of regular expressions (such as `xend-relocation-hosts-allow '^localhost\\.localdomain$'`). A host with a fully qualified domain name or IP address that matches these expressions are accepted.

在配置了这些参数之后，你必须重新启动主机，使红帽虚拟化系统接受你的新参数。

18.24. 翻译错误信息

你接收到下面的错误：

```
failed domain creation due to memory shortage, unable to balloon domain0
```

如果没有足够的可用内存，域将不能运行。Domain0 没有足够的空间来容纳新创建的客户机。你可以检查 `xend.log` 里关于这个错误的内容：

```
[2006-12-21] 20:33:31 xend 3198] DEBUG (balloon:133) Balloon: 558432 Kib
free; 0 to scrub; need 1048576; retries: 20
[2006-12-21] 20:33:31 xend. XendDomainInfo 3198] ERROR (XendDomainInfo:
202
Domain construction failed
```

You can check the amount of memory in use by domain0 by using the `xm list Domain0` command. If domain0 is not ballooned down, you can use the command "`xm mem-set Domain-0 NewMemSize`" to check memory.

你接收到下面的错误：

```
wrong kernel image: non-PAE kernel on a PAE
```

This message indicates that you are trying to run an unsupported guest kernel image on your Hypervisor. This happens when you try to boot a non-PAE paravirtual guest kernel on a RHEL 5.1 hypervisor. Red Hat Virtualization only supports guest kernels with PAE and 64bit architectures.

键入这个命令：

```
[root@smith]# xm create -c va base

Using config file "va-base"
Error: (22, 'invalid argument')
[2006-12-14 14:55:46 xend.XendDomainInfo 3874] ERRORs
(XendDomainInfo:202) Domain construction failed

Traceback (most recent call last)
File "/usr/lib/python2.4/site-packages/xen/xend/XendDomainInfo.py", line
195 in create vm.initDomain()
File " /usr/lib/python2.4/site-packages/xen/xend/XendDomainInfo.py", line
1363 in initDomain raise VmError(str(exn))
VmError: (22, 'Invalid argument')
[2006-12-14 14:55:46 xend.XendDomainInfo 3874] DEBUG (XenDomainInfo: 1449]
XendDlomainInfo.destroy: domin=1
[2006-12-14 14:55:46 xend.XendDomainInfo 3874] DEBUG (XenDomainInfo: 1457]
XendDlomainInfo.destroy:Domain(1)
```

如果你需要运行 32 位/非 PAE 内核，你将需要把客户机作为完全虚拟化的虚拟机运行。对于半虚拟化的客户机，如果你需要运行 32 位的 PAE 客户机，你必须具有 32 位的 PAE 监控程序。对于半虚拟化的客户机，如果你要运行 64 位的 PAE 客户机，你必须具有 64 位的 PAE 监控程序。对于完全虚拟化的客户机，你必须用 64 位的监控程序运行 64 位客户机。RHEL 5 i686 里的 32 位 PAE 监控程序只支持运行 32 位的并行虚拟化和 32 位的完全虚拟化的客户机操作系统。64 位监控程序只支持 64 位的并行虚拟化客户机。

This happens when you move the full virtualized HVM guest onto a RHEL 5.1 system. Your guest may fail to boot and you will see an error in the console screen. Check the PAE entry in your configuration file and ensure that `paе=1`. You should use a 32bit distribution.

你接收到下面的错误：

```
Unable to open a connection to the Xen hypervisor or daemon
```

当 `virt-manager` 程序不能启动时，会出现这个问题。当 `/etc/hosts` 配置文件里没有 `localhost` 条目时会产生这个错误。请确认配置文件里是否启用了 `localhost` 条目。下面是一个错误的 `localhost` 条目示例：

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
localhost.localdomain localhost
```

下面是一个正确的 `localhost` 条目示例：

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1 localhost.localdomain localhost
localhost.localdomain. localhost
```

你会接收到下面的错误（在 `xen-xend.log file` 文件里）：

```
Bridge xenbr1 does not exist!
```

This happens when the guest's bridge is incorrectly configured and this forces the Xen hotplug scripts to timeout. If you move configuration files between hosts, you must ensure that you update the guest configuration files to reflect network topology and configuration modifications. When you attempt to start a guest that has an incorrect or non-existent Xen bridge configuration, you will receive the following errors:

```
[root@trumble virt]# xm create r5b2-mysql01

Using config file " r5b2-mysql01"
Going to boot Red Hat Enterprise Linux Server (2.6.18.-1.2747 .el5xen)
kernel: /vmlinuz-2.6.18-12747.el5xen
initrd: /initrd-2.6.18-1.2747.el5xen.img
Error: Device 0 (vif) could not be connected. Hotplug scripts not working.
```

另外，`xend.log` 里会有下面的错误：

```
[2006-11-14 15:07:08 xend 3875] DEBUG (DevController:143) Waiting for
devices vif
[2006-11-14 15:07:08 xend 3875] DEBUG (DevController:149) Waiting for 0
[2006-11-14 15:07:08 xend 3875] DEBUG (DevController:464)
hotplugStatusCallback

/local/domain/0/backend/vif/2/0/hotplug-status

[2006-11-14 15:08:09 xend.XendDomainInfo 3875] DEBUG (XendDomainInfo:1449)
XendDomainInfo.destroy: domid=2
[2006-11-14 15:08:09 xend.XendDomainInfo 3875] DEBUG (XendDomainInfo:1457)
XendDomainInfo.destroyDomain(2)
```

```
[2006-11-14 15:07:08 xend 3875] DEBUG (DevController:464)
hotplugStatusCallback
```

```
/local/domain/0/backend/vif/2/0/hotplug-status
```

要解决这个问题，你必须编辑你的客户机配置文件，并修改 **vif** 条目。找到配置文件里的 **vif** 条目，假定你把 **xenbr0** 作为缺省网桥，正确的设置应该如下所示：

```
# vif = ['mac=00:16:3e:49:1d:11, bridge=xenbr0',]
```

你接收到这些 **python** 错误：

```
[root@python xen]# xm shutdown win2k3xen12
[root@python xen]# xm create win2k3xen12
```

```
Using config file "win2k3xen12".
```

```
/usr/lib64/python2.4/site-packages/xenxm/opts.py:520: Deprecation Warning:
Non ASCII character '\xc0' in file win2k3xen12 on line 1, but no encoding
declared; see http://www.python.org/peps/pep-0263.html for details
```

```
execfile (defconfig, globs, locs,)
Error: invalid syntax 9win2k3xen12, line1)
```

当遇到无效的（或不正确的）配置文件时，**Python** 生成这些错误。要解决这个问题，你必须更正不正确的配置文件，或者生成一个新的文件。

18.25. 关于故障解除的在线资源

- Red Hat Virtualization Center

```
http://www.openvirtualization.com
```

- Red Hat Enterprise Linux 5 Beta 2 文档

```
http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/index.html
```

- Libvirt API

```
http://www.libvirt.org
```

- virt-manager 项目主页

```
http://virt-manager.et.redhat.com
```

- Xen 社区中心

```
http://www.xensource.com/xen/xen/
```

- 虚拟化技术概述

■

■ <http://virt.kernelnewbies.org>

- Emerging Technologies Projects

■ <http://et.redhat.com>

第 19 章 其他的资源

要学习更多关于 Red Hat Virtualization 的知识，请参考下面的资源。

19.1. 有用的网站

- <http://www.cl.cam.ac.uk/research/srg/netos/xen/> – Xen™ 半虚拟化机器管理者的项目网站，Red Hat Virtualization 源于这个项目。这个站点维护了最新的 Xen 项目的二进制文件和源码，它也包含其他信息，如架构概述、文档和关于 Xen 及其相关技术的链接。
- <http://www.libvirt.org/> – 和主机 OS 的虚拟化框架进行交互的 **libvirt** 虚拟化 API 的官方网站。
- <http://virt-manager.et.redhat.com/> – **Virtual Machine Manager (virt-manager)**（一个管理虚拟机的图形化应用程序）的项目网站。

19.2. 安装的文档

- `/usr/share/doc/xen-<version-number>/` – This directory contains a wealth of information about the Xen para-virtualization hypervisor and associated management tools, including a look at various example configurations, hardware-specific information, and the current Xen upstream user documentation.
- `man virsh` and `/usr/share/doc/libvirt-<version-number>` – Contains subcommands and options for the `virsh` virtual machine management utility as well as comprehensive information about the **libvirt** virtualization library API.
- `/usr/share/doc/gnome-applet-vm-<version-number>` – Documentation for the GNOME graphical panel applet that monitors and manages locally-running virtual machines.
- `/usr/share/doc/libvirt-python-<version-number>` – Provides details on the Python bindings for the **libvirt** library. The `libvirt-python` package allows python developers to create programs that interface with the **libvirt** virtualization management library.
- `/usr/share/doc/python-virtinst-<version-number>` – Provides documentation on the `virt-install` command that helps in starting installations of Fedora and Red Hat Enterprise Linux related distributions inside of virtual machines.
- `/usr/share/doc/virt-manager-<version-number>` – Provides documentation on the Virtual Machine Manager, which provides a graphical tool for administering virtual machines.

附录 A. 实验 1

Xen 客户机安装

目标：安装 RHEL 3、4 或 5 和 Windows XP Xen 客户机。

先决条件：带有虚拟化组件的红帽企业 Linux 5.0 工作站。

在这个实验里，你将使用不同的虚拟化工具来配置并安装 RHEL 3、4 或 5 和 Windows XP Xen 客户机。

实验步骤 1：检查对 PAE 的支持

你必须检查你的系统是否支持 PAE。红帽虚拟化系统支持用基于 x86_64 或 ia64 CPU 的系统结构运行半虚拟化 (para-virtualized) 客户机。要运行 i386 客户机，系统需要带有 PAE 支持的 CPU。许多老式的笔记本 (尤其是基于 Pentium Mobile 或 Centrino 的笔记本) 不支持 PAE。

1. 要检查你的 CPU 是否支持 PAE，键入：

```
grep pae /proc/cpuinfo
```

2. 下面的输出显示了这个 CPU 支持 PAE。如果这个命令没有返回任何输出，说明这个 CPU 不支持 PAE。这个实验里所有的练习都要求带有 PAE 扩展的 i386 CPU 或者是 x86_64 和 ia64。

```
flags :
  fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat
  clflush dts acpi
  mmx fxsr sse sse2 ss tm pbe nx up est tm2
```

实验步骤 2：用 virt-install 安装 RHEL5 Beta 2 Xen 半虚拟化客户机。

在这个实验里，你必须使用 virt-install 安装红帽企业 Linux 5 Beta 2 的 Xen 客户机。

1. 要安装红帽企业 Linux 5 Beta 2 的 Xen 客户机，在命令行下键入：**virt-install**。
2. 当你被询问是否安装完全虚拟化的客户机时，输入：**no**。
3. 输入 **rhel5b2-pv1** 作为你的虚拟机名称。
4. 输入 **500** 作为分配的内存数量。
5. 输入 **/xen/rhel5b2-pv1.img** 作为你的磁盘 (客户机映像)。
6. 输入 **6** 作为你的磁盘 (客户机映像) 大小。
7. 输入 **yes** 来启用图形化支持。
8. 输入 **nfs:server:/path/to/rhel5b2** 作为安装位置。
9. 安装开始了。象平常一样进行安装。
10. 在安装结束后，键入 **/etc/xen/rhel5b2-pv1**，并进行下面的修改：
#vnc=1#vncunused=1sdl=1

11. 使用文本编辑器来修改 `/etc/inittab`，并把下面的内容附加到文件里：`init 5.#id:3:initdefault:id:5:initdefault:`

实验步骤 3：用 `virt-manager` 安装 RHEL5 Beta 2 Xen 半虚拟化客户机。

在这个实验里，你将用 `virt-manager` 来安装红帽企业 Linux 5 Beta 2 的 Xen 半虚拟化客户机。

1. 要安装红帽企业 Linux 5 Beta 2 的 Xen 半虚拟化客户机，在命令行提示下输入：`virt-manager`。
2. 在 Open Connection 窗口，选择 Local Xen host，然后点击 **Connect**。
3. Start Red Hat's Virtual Machine Manager application, and from the **File** menu, click on **New**.
4. 点击 **Forward**。
5. 键入 `rhe15b2-pv2` 作为你的系统名，然后点击 **Forward**。
6. 选择 **Paravirtualized**，并点击 **Forward**。
7. 键入 `nfs:server:/path/to/rhe15b2` 作为安装介质的 URL，然后点击 **Forward**。
8. 选择 **Simple File**，键入 `/xen/rhe15b2-pv2.img` 作为文件位置。选择 6000 MB，然后点击 **Forward**。
9. 选择 500 作为你的虚拟机的启动和最大内存，然后点击 **Forward**。
10. 点击 **Finish**。

虚拟机控制台窗口将出现。象往常一样进行并结束安装。

实验步骤 4：检查对 Intel-VT 或 AMD-V 的支持。

在这个实验里，你必须检查你的系统是否支持 Intel-VT 或 AMD-V 硬件。要成功安装完全虚拟化的客户机操作系统，你的系统必须支持启用 Intel-VT 或 AMD-V 的 CPU。红帽虚拟化系统合并了一个通用的 HVM 层来支持这些 CPU。

1. To determine if your CPU has Intel-VT or AMD-V support, type the following command:
`egrep -e 'vmx|svm' /proc/cpuinfo`
2. 下面的输出表示 CPU 支持 Intel-VT：

```
.flags :
    fpu tsc msr pae mce cx8 apic mtrr mca cmov pat clflush dts acpi
mmx fxsr sse
    sse2 ss ht tm pbe constant_tsc pni monitor vmx est tm2 xtpr
```

如果这个命令没有任何输出，则表示 CPU 不支持 Intel-VT 或 AMD-V。

3. 要知道你的 CPU 是否支持 Intel-VT 或 AMD-V，键入下面的命令：

```
cat /sys/hypervisor/properties/capabilities
```

4. The following output shows that Intel-VT support has been enabled in the BIOS. If the command returns nothing, then go into the BIOS Setup Utility and look for a setting related to

'Virtualization', i.e. 'Intel(R) Virtualization Technology' under 'CPU' section on a IBM T60p. Enable and save the setting and do a power off to take effect.

```
xen-3.0-x86_32p hvm-3.0-x86_32 hvm-3.0-x86_32p
```

实验步骤 5：用 `virt-install` 安装 RHEL5 Beta 2 Xen 完全虚拟化客户机。

在这个实验里，你将用 `virt-install` 安装红帽企业 Linux 5 Beta 2 Xen 完全虚拟化客户机。

1. 要安装红帽企业 Linux 5 Beta 2 的 Xen 客户机，在命令行下键入：`virt-install`。
2. 当被提示安装完全虚拟化客户机时，键入 `yes`。
3. 键入 `rhe15b2-pv2` 作为你的虚拟机名。
4. 键入 500 作为分配的内存数量。
5. 键入 `/xen/rhe15b2-fv1.img` 作为你的磁盘（客户机映像）。
6. 输入 6 作为你的磁盘（客户机映像）大小。
7. 键入 `yes` 来启用图形化支持。
8. 键入 `/dev/cdrom` 作为虚拟 CD 映像。
9. The VNC viewer appears within the installation window. If there is an error message that says “main: Unable to connect to host: Connection refused (111)”, then type the following command to proceed: `vncviewer localhost:5900`. VNC port 5900 refers to the first Xen guest that is running on VNC. If it doesn't work, you might need to use 5901, 5902, etc.

安装开始了。象平常一样进行安装。

实验步骤 6：用 `virt-manager` 安装 RHEL5 Beta 2 Xen 完全虚拟化客户机。

在这个实验里，你将使用 `virt-manager` 安装红帽企业 Linux 5 Beta 2 Xen 完全虚拟化客户机：

1. 要安装红帽企业 Linux 5 Beta 2 的 Xen 半虚拟化客户机，在命令行提示下输入：`virt-manager`。
2. 在 **Open Connection** 窗口，选择 Local Xen host，并点击 **Connect**。
3. Start Red Hat's Virtual Machine Monitor application, and from the **File** menu, click on **New**.
4. 点击 **Forward**。
5. 键入 `rhe15b2-fv2` 作为你的系统名，然后点击 **Forward**。
6. 选择 Fully virtualized，并点击 **Forward**。
7. 指定 CD-ROM 或 DVD，并输入安装介质的路径。如果你将从 ISO 映像安装的话，指定 ISO 映像的位置。点击 **Forward**。
8. 选择 Simple File，键入 `/xen/rhe15b2-fv2.img` 作为文件的位置。指定 6000 MB，并点击 **Forward**。

9. 选择 500 作为你的虚拟机的启动和最大内存，然后点击 **Forward**。

10. 点击 **Finish** 。

11. 然后虚拟机控制台窗口将出现。

如往常一样进行并结束安装。

实验步骤 7：用 **virt-manager** 安装 RHEL3 Xen 完全虚拟化客户机。

在这个实验里，你将用 **virt-manager** 安装红帽企业 Linux 3 Xen 客户机。

1. 在这里你可以采用和实验步骤 6 相同的说明。

实验步骤 8：用 **virt-manager** 安装 RHEL4 Xen 完全虚拟化客户机。

在这个实验里，你将用 **virt-manager** 安装红帽企业 Linux 4 Xen 客户机。

1. 在这里你可以采用和实验步骤 6 相同的说明。

实验步骤 9：使用 **virt-manager** 安装 Windows XP Xen 完全虚拟化客户机。

在这个实验里，你将用 **virt-manager** 安装一个完全虚拟化的 Windows XP Xen 客户机。

1. 要在 Windows XP 主机上安装红帽企业 Linux 5，在命令行下输入：**virt-manager**。

2. 在 **Open Connection** 窗口里，选择 **Local Xen host**，然后点击 **Connect**。

3. Start Red Hat's Virtual Machine Manager application, and from the **File** menu click on **New**.

4. 点击 **Forward**。

5. 键入 **winxp** 作为系统名，然后点击 **Forward**。

6. 选择 **Fully virtualized**，并点击 **Forward**。

7. 指定 CD-ROM 或 DVD，然后输入安装介质的路径。如果你将从 ISO 映像进行安装，指定 ISO 映像的位置。点击 **Forward**。

8. 选择 **Simple File**，键入 **/xen/winxp.img** 作为文件的位置。指定 6000 MB，并点击 **Forward**。

9. 选择 1024 作为虚拟机的启动和最大的内存数量，然后选择 2 作为 VCPU 的数量。点击 **Forward** 。

10. 点击 **Finish**。

11. 虚拟机控制台窗口将出现。象往常一样进行并结束安装。

12. 选择把 **C:** 分区格式化为 **FAT** 文件系统格式。红帽企业 Linux 5 没有 **NTFS** 内核模块。如果你想把分区格式化为 **NTFS** 格式，挂载或写入文件到 Xen 客户机映像里可能不是件简单的事。

13. 在你第一次重新启动系统之后，编辑 **winxp** 客户机映像：**losetup /dev/loop0 /xen/winxp.imgkpartx -av /dev/loop0mount /dev/mapper/loop0p1 /mntcp -prv \$WINDOWS/i386 /mnt/**。这可以修复你在后面的 Windows 安装部分可能遇到的一个问题。

14. 键入 `xm create -c winxp/` 手工重新启动 Xen 客户机。
15. 在虚拟机管理者窗口，选择 `winxp` Xen 客户机并点击 **Open**。
16. 虚拟机控制台窗口会出现。象平常一样进行并结束安装。
17. Whenever a 'Files Needed' dialog box appears, change the path `GLOBALROOT\DEVICE\CDROM0\I386` to `C:\I386`. Depending on your installation, you may or may not see this problem. You may be prompted for missing files during the installation. Changing the path to `C:\I386` should compensate for this problem.
18. 如果 Xen 客户机控制台不动了，点击 `shutdown`，进行如下的修改：
`/etc/xen/winxp:#vnc=1#vncunused=1sd1=1#vcpus=2`。
19. 重复步骤 14 并象平常一样进行安装。

附录 B. 实验 2

Live 移植

目标：在两台主机之间配置并执行 live 移植。

先决条件：两台装有附带虚拟化平台的红帽企业 Linux 5.0 Beta 2 的工作站，其中一台装有 Fedora Core 6 Xen 客户机。

在这个实验里，你将为移植进行配置并在这两个主机之间执行移植。

说明：在你开始之前

在这个实验里，你将需要两台虚拟化主机：一个 Xen 客户机和一个用于共享存储空间的主机。你必须用 UTP 电缆连接这两台主机。其中一台用 NFS 导出共享的存储空间。你必须配置这两台机器来成功地进行移植。Xen 客户机存放在共享的存储空间里。在 Xen 客户机上，你应该安装一个 streaming 服务器。当 live 移植在两台主机间进行时，你必须确保这个 streaming 服务器在 Xen 客户机上不间断地运行。对于实验 2，这两台虚拟化主机被称为 **host1** 和 **host2**。

步骤 1：配置 xend（两台 Xen 主机）

在本实验的过程中，你将配置 xend 来作为 HTTP 服务器和 relocation 服务器启动。缺省情况下，xend 守护进程不初始化 HTTP 服务器。它启动 UNIX 域套接字管理服务器（xm）并与 xend 进行通信。要启用跨机器的 live 移植，你必须对它进行配置：

1. 备份 **xend-config.sxp** 文件：

```
cp -pr /etc/xen/xend-config.sxp /etc/xen/xend-config.sxp.default
```

2. 编辑 **/etc/xen/xend-config.sxp** 并进行如下修改：

```

#(xend-unix-server yes)(xend-relocation-server
    yes)(xend-relocation-port 8002)(xend-relocation-address
    '')(xend-relocation-hosts-allow '')#(xend-relocation-hosts-allow
'^localhost$
^localhost\\.localdomain$')
```

3. 重启 **xend:service** 并 **xend restart**。

步骤 2：通过 NFS 导出共享存储空间

在这个实验的过程中，你将配置 NFS 并用它来导出一个共享存储空间。

1. 编辑 **/etc/exports** 并包含如下的一行：**/xen *(rw, sync, no_root_squash)/**
2. 保存 **/etc/exports** 并重新启动 NFS 服务器。确保用 **default:service nfs startchkconfig nfs on** 启动 NFS 服务器。
3. 在 **host1** 上启动 NFS 服务器后，我们就可以挂载它：**host2:mount host1:/xen**。
4. 现在在 **host1** 上启动 Xen 客户机并选择 **fc6-pv1**（或实验 1 里的 **fc6-pv2**）：

```
xm create -c fc6-pv1
```

步骤 3：安装 Xen 客户机的 streaming 服务器

在这个步骤里，你将安装用于演示目的的 streaming 服务器：gnump3d。你将选择 gnump3d，因为它支持 OGG vorbis 文件且易于安装、配置和修改。

1. 从 <http://www.gnump3d.org/> 下载 `gnump3d-2.9.9.9.tar.bz2` tarball。在 `gnump3d-2.9.9.9/` 目录里解压这个 tarball 文件、编译并安装：`gnump3d application:tar xvjf gnump3d-2.9.9.9.tar.bz2cd gnump3d-2.9.9.9/make install`
2. Create a `/home/mp3` directory and copy TruthHappens.ogg from Red Hat's Truth Happens page to `mkdir /home/mp3wget -c http://www.redhat.com/v/ogg/TruthHappens.ogg`
3. 输入下面的命令来启动 streaming 服务器

```
command:gnump3d
```

4. 在两台 Xen 主机的任何一台上，运行 Movie Player。如果没有安装，在运行 Movie Player 之前安装 `totem` 和 `iso-codecs` 软件包。点击 Applications，然后是 Sound & Video，最后选择 Movie Player。
5. 点击 Movie，然后 Open Location，输入 `http://guest:8888/TruthHappens.ogg`。

步骤 4：执行 live 移植

1. 在两台 Xen 主机的其中一台上运行 `TruthHappens.ogg` 文件。
2. 执行从 `host1` 到 `host2` 的 live 移植：

```
xm migrate -live fc6-pv1 host2
```

3. 用下面的命令在两台 Xen 主机上打开多个窗口终端：

```
watch -n1 xm list
```

4. 观察 live 移植。注意完成这个移植需要多少时间。

挑战性的步骤：在 Xen 客户机里配置 VNC 服务器

If time permits, from within the Xen guest, configure the VNC server to initiate when `gdm` starts up. Run VNC viewer and connect to the Xen guest. Play with the Xen guest when the live migration occurs. Attempt to pause/resume, and save/restore the Xen guest and observe what happens to the VNC viewer. If you connect to the VNC viewer via `localhost:590x`, and do a live migration, you won't be able to connect to the VNC viewer again when it dies. This is a known bug.

附录 C. REVISION HISTORY

修订 2.0-11.400 Rebuild with publican 4.0.0	2013-10-31	Rüdiger Landmann
修订 2.0-11 Rebuild for Publican 3.0	2012-07-18	Anthony Towns
修订 5.1.0-10 Resolves: #245684 Content Updates	Fri Aug 3 2007	Michael Hideo Smith