第 1 章 红帽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。

你可以以两种方式来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器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。

hypervisor（红帽的虚拟机监控器）是一个虚拟化平台，它允许多个操作系统在单一的主机里的完全虚拟化环境里同时运行。客户机（guest）是主机或主操作系统之外的运行在虚拟机上的操作系统（OS）。

在红帽虚拟化系统里，每个客户机的内存（**memory**）都来自于一片主机的物理内存。对于半虚拟化的客户机，你可以设置初始内存和虚拟机的最大值。你也可以在运行时增加（或删除）虚拟机的物理内存，只要不超过你所指定的最大值。这个过程被称为 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 地址从一个有着一千六百万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。

客户机可以用三种身份（**identity)**中的任何一种来识别：域名（domain-name）、identity（domain-id）或 UUID。domain-name 是一个对应于客户机配置文件的文本串。domain-name 用来启动客户机，当客户机运行时，相同的名字被用来进行识别和控制。domain-id 是唯一的，非持久的号码，它被分配给活动的域并被用来进行识别和控制。UUID 是持久的、唯一的识别符，它在客户机的配置文件里进行控制并确保客户机可以被系统管理工具所识别。在客户机运行时，它是可见的。当客户机第一次安装时，新的 UUID 被系统工具自动分配给每个客户机。

第 2 章 操作系统支持

红帽虚拟化系统的半虚拟化模式允许你在可能很难虚拟化的体系结构（如基于 x86 的系统）上应用高性能的虚拟化。要在你的操作系统里部署半虚拟化，你需要访问半虚拟化客户机内核，这可以从不同的

红帽 distro（例如，RHEL 4.0、RHEL 5.0 等）里获得。虽然你的操作系统内核必须支持红帽虚拟化系统，但你并不需要修改用户的应用程序或库文件。

如果你有 Intel VT 和 AMD SVM CPU 硬件，红帽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。红帽虚拟化系统支持：

- 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。
- 作为应用 Intel Vanderpool 或 AMD Pacifica 技术的未经修改的客户机操作系统的 Microsoft Windows。

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 mmtext 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

红帽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）处理器运行红帽虚拟化系统：

- 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  (2.618-4.elxen)

root  (hd0, 0)

kernel   /xen.gz-2.618-4-el5 mem=32G

module   /vmlinuz -2.618-4.el5xen ro root=LABEL=/

module   /initrd-2.618-4.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

mmtext 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 红帽虚拟化系统的系统要求

下面列出的项目是红帽虚拟化系统所要求的:

- 正常运行的红帽 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。

第 5 章 引导系统

在安装了红帽虚拟化组件后，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。而且在你启动红帽虚拟化系统之前，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.sxp 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（或 GRUB）是一个让用户在系统引导时选择安装的操作系统或内核的程序。它也允许用户把参数传入内核。GRUB 的配置文件（/boot/grub/grub.conf）用来创建 GRUB 的菜单里界面里的操作系统列表。当你安装 kernel-xen RPM 软件包时，安装后（post）脚本把 kernel-xen 条目加入到 GRUB 的配置文件里。你可以手工编辑 grub.conf 并启用下面的 GRUB 参数：

```
title Red Hat Enterprise Linux Server (2.618-3.el5xen)

root   (hd0; 0)

kernel  /xen.gz.-2.618-3.el5

module  /vmlinuz-2.6.18-3.el5xen ro root=/dev/VolGroup00/LogVol00  rhgb quiet

module  /initrd-2.618-3. 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 (2.618-3.el5xen)

root   (hd0; 0)
```

```
kernel  /xen.gz.-2.618-3.el5 dom0_mem=800M

module  /vmlinuz-2.6.18-3.el5xen ro root=/dev/VolGroup00/LogVol00  rhgb quiet

module  /initrd-2.618-3. 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。 -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
```

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

第 9 章 配置文件

红帽虚拟化系统的配置文件包含下面的标准变量。这些文件里的配置项必须用引号（"）括起来。这些配置文件位于 /etc/xen 目录里。

项目	描述
pae	指定物理地址扩展配置数据。
apic	指定高级可编程的中断控制器配置数据。
memory	指定以 megabyte 为单位的内存大小
vcpus	指定虚拟 CPU 的数量。
console	指定导出域控制台的端口号。
nic	指定虚拟网络接口的数量。
vif	列出随机分配的 MAC 地址和用于域网络地址的网桥。
disk	列出导出至域的块设备并以只读权限导出物理设备到域。
dhcp	用 DHCP 启用网络。
netmask	指定配置的 IP 掩码。
gateway	指定配置的 IP 网关。
acpi	指定高级配置电源接口的配置数据。

表 **9.1.** 红帽虚拟化系统配置文件

第 10 章 管理 CPU

红帽虚拟化系统允许某个域的虚拟化 CPU 与一个或多个主机 CPU 相关联。这可以用来在一个或多个客户机上分配实际的资源。当应用双核、超线程或其他高级的 CPU 技术时，这种方法可以优化红帽虚拟化系统对处理器资源的使用。如果你正在运行对 I/O 需求很高的任务，分配单独的线程或整个内核来运行 domain0 是一个典型的更好的办法。红帽虚拟化系统的信用调度程序（credit scheduler）自动地在物理 CPU 之间调整虚拟 CPU 的使用来最大限度地利用系统资源。只要虚拟 CPU 固定在某个物理 CPU 上，红帽虚拟化系统就允许信用调度程序按需要移动它。

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。

当域在移植时，它的 MAC 和 IP 地址也随之转移。只有具有相同的 layer-2 网络和子网的虚拟机才能成功移植。如果目的节点是在不同的子网里，管理员必须手工地配置在 domain0 的远程节点里的合适的 EtherIP 或 IP 管道。xend 守护进程停止域，然后把它复制到新的节点并重新启动它。除了 localhost（请参阅 /etc/xend-config.sxp 文件），红帽虚拟化系统的 RPM 软件包不启用来自其他任何主机的移植。如果目的主机要接受远程主机的移植请求，你必须修改目的主机的 xen-relocation-hosts-allow 参数。因为没有验证机制，请小心配置允许哪些主机进行移植。

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。缺省的红帽虚拟化系统的 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]
```

你可能需要重新连接到新机器的域控制台。你可以使用 xm console 命令来进行重新连接。

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。

每个域网络接口都使用点到点链接来连接到 dom0 里的虚拟网络接口。这些设备以 vif、<domid> 和 <vifid> 来命名。vif1.0 是域 1 里的第一个接口；vif3.1 是域 3 里的第二个接口。

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。

红帽虚拟化系统的虚拟网络是由两个脚本控制的：network-bridge 和 vif-bridge。当某些事件发生时，**xend** 调用这些脚本。可以给这些脚本传入参数来提供额外的上下文信息。这些脚本位于 /etc/xen/scripts 目录。你可以用 /etc/xen 目录下的 xend-config.sxp 配置文件来改变这些脚本的属性。

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红帽虚拟化系统，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

当在你的公司基础架构里部署红帽虚拟化系统时，你必须确保 domain0 不被破解。Domain0 是处理系统管理的专用域。如果 domain0 不安全，所有系统里的其他域都是易受攻击的。在系统里集成红帽虚拟化系统时，你应该知道几个实施安全性的方法。与公司里的其他同事一起，你应该创建一个部署计划，它包含操作规格和将在红帽虚拟化系统里运行的服务的，以及支持这些服务所需要的东西。下面是编写部署计划时需要考虑的一些安全性问题：

- 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>
```

这里的 <name> 是监控程序所在的机器名。如果你想初始化一个只读连接，把 —readonly 附加在命令后面。

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** 创建的客户机，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]
```

这个命令输出域信息（XML 格式）到 stdout 。如果你把这些数据保存到文件里，你可以使用 create 选项来重新创建虚拟机。

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]
```

当某个域处于挂起状态时，它仍旧消耗系统内存。挂起时不会有磁盘或网络 I/O。这个操作是立时生效的，虚拟机必须用 resume 选项进行重新启动。

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]
```

这个命令停止你指定的虚拟机并把数据保存到文件里，这个过程所需的时间依据虚拟机使用的内存数量而有所不同。你可以用 restore 选项 来恢复虚拟机的状态。

15.7. 恢复虚拟机

你可以使用 **virsh** 来恢复之前用 virsh save 选项所保存的虚拟机。

```
virsh restore [filename]
```

这个命令重新启动了保存的虚拟机，这会需要一段时间。虚拟机的名字和 UUID 都会被保留，但会分配一个新的 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]
```

这个命令执行一个立时的 ungraceful 关闭并停止任何客户域会话（这有可能导致仍被虚拟机使用的文件系统的崩溃）。只有当虚拟机操作系统不响应时，你才应该使用 destroy 选项。对于半虚拟机，你应该使用 shutdown 选项。

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

你可以用 **virsh** 来修改虚拟机的 CPU 数量。

```
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

你可以用 **virsh** 来修改某个域的内存分配:

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

你必须指定 [count] （以 KB 为单位）。注意，新的值不能超过你创建虚拟机时指定的数量。低于 64MB 的值可能没法工作。你可以根据需要调整虚拟机的内存。

15.21. 配置最大内存数量

你可以使用 **virsh** 来修改虚拟机的最大内存数量：

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

你必须指定 [count]（以 KB 为单位）。注意，新的值不能超过你创建虚拟机时指定的数量。低于 64MB 的值可能没法工作。最大内存数量不会影响当前虚拟机的使用（除非设定了一个更低的值，这会紧缩内存的使用）。

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：

项目	描述
console-limit	决定控制台服务器的内存缓冲限制，以域为基础进行分配。
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
xend-relocation-address	指定虚拟机允许系统移植的地址
xend-address	指定域套接字服务器绑定的地址

表 **16.1.** 红帽虚拟化系统的 **xend** 配置参数

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
```

下面的输出显示了守护进程的状态。

第 **17** 章 使用 **xm** 管理虚拟机

xm 应用程序是一个可靠的管理工具，它允许你配置你的红帽虚拟化环境。使用 **xm** 的前提条件是，你必须确保 **xend** 守护进程在系统里运行。

17.1. xm 配置文件

xmdomain.cfg 文件包含你必须修改的操作参数，它位于 etc/xen 目录。下面是一些你可以在 xmdomain.cfg 文件里启用或禁止的参数：

项目	描述
kernel	决定内核映像的全限定路径
ramdisk	决定用于初始 ramdisk 的 initrd 的全限定路径
memory	决定在域启动时所分配的内存数量（以 MB 为单位）
name	决定域的唯一名称
root	决定域的根设备
nic	决定域的网络接口卡的数量（缺省为 1）
disk	决定设备块队列的模式 — 三种模式分别是： · mode - 设备访问模式 · backend-dev - 导出至客户机域的后台域 · frontend-dev - 决定设备怎样在客户机域里出现
vif	决定虚拟接口队列的模式（每个模式代表一组 name=value 操作）。
builder	决定构造域的建立者（缺省为 Linux）
cpu	决定域启动时所用的 CPU 序号。0 表示第一个 CPU，1 表示第二个，诸如此类（缺省为 -1）。
cpus	决定域的 VCPU 上的哪些 CPU 是可执行的
extra	决定在内核参数行最后所附加的其他信息
nfs_server	决定用于根设备的 NFS 服务器的 IP 地址
nfs_root	决定 NFS 服务器的根目录的全限定路径
vcpus	决定分配给域的虚拟 CPU 数量（缺省为 1）
on_shutdown	设定域关闭（shutdown）的参数来触发 DomU 内部的 graceful 关闭（或 **xm** shutdown）
on_reboot	设定域关闭的参数来触发 DomU 内部的 graceful 重启（或 **xm** reboot）
on_crash	指定触发 DomU 崩溃的域关闭参数。

表 **17.1. xmdomain.cfg** 配置文件

17.1.1. 配置 vfb

vfb 是一个被定义为 'stanza' 的虚拟帧缓冲。stanza 代表一组 name = value 的操作，它被集成在 xmdomain.cfg.5 文件里，且用逗号隔开。配置文件里的 **vfb** 条目和下面相似：

```
vfb = [ "stanza" ]  "name1=value1, name2=value2, "
```

通过合并 Table 16.2 里显示的选项，你可以进一步配置你的 **vfb** 环境：

项目	描述
type	vnc type 选项初始化一个连接至外部 VNC viewer 的 VNC 服务器会话。sdl 选项初始化内部的 viewer。
vncdisplay	决定所使用的 VNC display 号码（缺省为域的 ID 值）。VNC 服务器侦听的端口是 5900 + display 号码。
vnclisten	VNC 服务器的侦听地址（缺省为 127.0.0.1）。
vncunused	数字值，如果非零的话，使 VNC 服务器侦听于 5900 之上的第一个未使用的端口。
vncpasswd	覆盖 Xend 所配置的缺省密码。
display	启用内部 viewer 使用的显示器（缺省为环境变量 DISPLAY）。
xauthority	启用内部 viewer 使用的授权文件（缺省为环境变量 XAUTHORITY）。

表 **17.2. vfb** 配置选项

17.2. 用 xm 创建和管理域

你可以使用 **xm** 程序来创建和管理域。

17.2.1. 连接至域

你可以使用 **xm** 来连接至域或虚拟机：

```
xm console domain-id
```

这使控制台附加到 domain-id 的文本控制台。

17.2.2. 创建域

你可以使用 xm 来创建域：

```
xm create domain001 [-c]
```

这创建了一个名为 domain001 的域，域文件位于 /etc/xen/ 目录。[-c] 选项允许你连接至文本控制台来协助解除故障。

17.2.3. 保存域

你可以使用 **xm** 来保存域：

```
xm save [domain-id] [statefile]
```

17.2.4. 终止域

你可以使用 **xm** 来终止域：

```
xm destroy [domain-id]
```

这会立即终止 domain-id。如果你想用另外一种方法来安全地终止这个会话，你可以使用 shutdown 参数。

17.2.5. 关闭域

你可以使用 **xm** 来关闭任何域：

```
xm shutdown [domain-id] [ -a | -w ]
```

[-a] 选项会关闭系统里的所有域。 [-w] 选项用于等待域的完全关闭。

17.2.6. 恢复域

你可以用 **xm** 来恢复以前保存的域。

```
xm restore [state-file]
```

17.2.7. 挂起域

你可以用 **xm** 来挂起域：

```
xm suspend [domain-id]
```

17.2.8. 使域继续运行

你可以用 **xm** 来使之前被暂停的域继续运行：

```
xm resume [domain-id]
```

17.2.9. 重新启动域

你可以使用 **xm** 来重启域：

```
xm reboot [domain-id] [ -a | -w ]
```

[-a] 选项将重启系统里所有的域。[-w] 选项用来等待某个域的重启完成。通过修改 xmdomain.cfg 文件里的 on_boot 参数，你可以控制域重新启动的行为。

17.2.10. 域的重命名

你可以用 xm 来给域分配一个新的名字：

```
xm rename [domain-name] [new domain-name]
```

域的重命名将保持相同的设置（相同的硬盘、内存等等）。

17.2.11. 暂停域

你可以使用 **xm** 来暂停域：

```
xm pause [domain-id]
```

17.2.12. 使域恢复运行

你可以用 **xm** 来使域恢复运行：

```
xm unpause [domain-id]
```

这使得监控程序（hypervisor）可以调度域。

17.2.13. 转换域名到 Domain ID

你可以使用 xm 来把域名转换为 domain ID：

```
xm domid [domain-name]
```

17.2.14. 把 Domain ID 转换为域名

你可以使用 **xm** 把 domain ID 转换为域名：

```
xm domname [domain-id]
```

17.2.15. 配置内存分配

你可以使用 **xm** 来修改域的内存分配：

```
xm mem-set [domain-id] [count]
```

注意

分配给域的内存不能够超过你第一次创建域时指定的最大数量。

17.2.16. 配置最大内存数量

你可以用 **xm** 来修改域的最大内存数量：

```
xm mem-max [domain-id] [count]
```

你必须指定 [count]（以 MB 为单位）。

17.2.17. 配置 VCPU 数量

你可以使用 **xm** 来修改域的 VCPU 数量：

```
xm vcpu-set [domain-id] [count]
```

你必须指定 [count]（以 MB 为单位）。

注意

分配给域的内存不能够超过你第一次创建域时指定的最大数量。

17.2.18. 固定（Pinning）VCPU

你可以用 xm 来固定某个 VCPU：

```
xm vcpu-pin [domain-id] [vcpu] [cpus]
```

这里的 [vcpu] 是你希望指定的 VCPU，[cpus] 是目标。固定（Pinning）保证了某些 VCPU 只能运行在某些 CPU 上。

17.2.19. 移植域

你可以使用 **xm** 来移植域：

```
xm migrate [domain-id] [host] [options]
```

这里的 [domain-id] 是你要移植的域，[host] 是目标。[options] 包括 ——live （或 -1）live 移植，或 ——resource （或 -r）指定移植的最大速度（以 Mbs 为单位）。

要确保成功地移植，你必须保证 xend 守护进程运行在所有的主机域上。所有的主机必须也运行红帽 RHEL 5.0+ 并打开用于移植的 TCP 端口来接受源主机的连接。

17.3. 监控和诊断

17.3.1. 执行核心转储

你可以用**xm** 来执行现存虚拟机的内存转储。

```
xm dump-core [-C] [domain-id]
```

这个命令把虚拟机的内存转储到 /var/xen/dump/ 目录里的 xendump 文件里。你可以用 -C 选项来终止虚拟机。

17.3.2. 实时地对域进行监控

你可以使用 **xm** 实时地监控域和主机：

```
xm top [domain-id]
```

17.3.3. 显示域状态

你可以使用 xm 来显示一个或多个域的活动状态：

```
xm list [domain-id] [ ——long  |  ——label]
```

你可以用名称来指定特定的域。[——long] 选项提供你所指定的域的更详细的信息。[——label] 选项添加了一个字段来显示 label 状态。输出如下：

```
Name                ID           Mem(MiB)       VCPUs       State       Time       Label

————————————————

Domain0             0                 927               8              r————    204.9

INACTIVE

Domain202        1                927               8             s————

205.0/command ACTIVE

DomainQ/A       2                927               8              b————

INACTIVE

Domain9600      3                927               8             c————      2051

ACTIVE
```

每个 VCPU 有六个域状态：

状态	描述
running	列出当前在某个 CPU 上活动的域
blocked	列出被阻塞的域（当 vcpu 在等待外部事件的发生时，域将被阻塞）
paused	列出被挂起的域
shutdown	列出正被关闭的域
shutoff	列出已被完全关闭的域

状态	描述
crashed	列出崩溃的域
inactive	列出是不活动实例的域
——all	列出是活动或不活动的 vcpu 实例的域

表 **17.3.** 域状态

17.4. 显示运行时间

你可以用 **xm** 来显示运行时间：

```
xm uptime [domain-id]
```

输出如下：

```
Name                          ID                Uptime

Domain0                       0                 4:45:02

Domain202                     1                 3:32:00

Domain9600                    2                 0:09:14

DomainR&D                     3                 2:21:41
```

17.5. 显示 VCPU 信息

你可以用 **xm** 来显示域的 CPU 信息：

```
xm vcpu-list [domain-id]
```

你必须指定你要列出哪些 vcpu。如果没有指定，所有域的 vcpu 都将被列出。

17.6. 显示域信息

你可以使用 **xm** 来显示主机域信息：

```
xm info
```

输出如下：

```
host                                   :  redhat83-157brisbane.redhat.com
release                                :  2.6.18-1.2741.el5xen
version                                :  #1 SMP Mon Oct 21 14:57:21  EDT  2006
machine                                :  x86_64
nr_cpus                                :  8
nr_nodes                               :  1
sockets_per_node                       :  2
cores_per_socket                       :  2
threads_per_core                       :  2
cpu_mhz                                :  2992
hw_caps                                :  bfeebbef:20100000:00000000:00000000
total_memememory                       :  1022
free_memory                            :  68
xen_major                              :  3
xen_minor                              :  0
xen_extra                              :  -unstable
```

```
xen_caps                               :  xen-3.0-x86_84
xen_pagesize                           :  4096
platform_params                        :  virt_start=0xffff880000000000000000000
xen_changeset                          :  unavailable
cc_compiler                            :  gcc compiler version 411 200060928
cc_compile_by                          :  brewbuilder
cc_compile_domain                      :  build.redhat.com
cc_compile_date                        :  Mon Oct 2 700 EDT 2006
xend_config_format                     :  2
```

17.7. 显示 TPM 设备

你可以使用 **xm** 来显示虚拟 TPM 设备：

```
xm vtpm-list [domain-id] [——long]
```

[——long] 选项提供了你所指定的域的更详细的信息。

17.8. 显示 xend 日志

你可以使用 **xm** 来显示 xend 日志的内容：

```
xm log
```

输出显示了 xend 的活动记录。

17.9. 显示消息缓冲（Message Buffer）

你可以使用 **xm** 来查看 xend 的消息缓冲：

```
xm dmesg
```

输出显示了 xend 消息缓冲的内容。

17.10. 显示 ACM 状态信息

你可以使用 **xm** 来显示监控程序（hypervisor）的 ACM 状态信息：

```
xm dumppolicy [policybin]
```

17.11. 显示 Vnet

你可以用 xm 来查看虚拟网络设备：

```
xm vnet-list [ -l  |  ——long]
```

输出如下：

```
List Vnets

-l,  ——long          List Vnets as SXP
```

17.12. 显示虚拟块设备

你可以用 **xm** 来查看域的虚拟块设备：

```
xm block-list [domain-id] [ ——long]
```

输出将显示你所指定的域的块设备。

17.13. 显示虚拟网络接口

你可以使用 **xm** 查看域的虚拟网络设备：

```
xm network-list [domain-id]  [ ——long]
```

输出将显示你所指定的域的网络接口。

17.14. 创建新的虚拟网络设备

你可以使用 **xm** 来创建一个新的虚拟网络设备：

```
xm network-attach [domain-id] [script=scriptname] [ip=ipaddr] [mac-macaddr]

[bridge=bridge-name] [backend-bedomain-id]
```

下面是五个可选参数：

参数	描述
[script=scriptname]	使用指定的脚本来启动网络
[ip=ipaddr]	把指定的脚本名传给适配器
[mac-macaddr]	域以太设备的 MAC 地址
[bridge-bridgename]	附加 vif 的设备的名称
[backend=bedomain-id]	后台的域 id

表 **17.4.** 参数

17.15. 终止虚拟网络设备

你可以使用 **xm** 来终止一个现存的虚拟网络设备：

```
xm network-detach [domain-id] [DevID]
```

这个命令终止了你所指定的虚拟网络设备。

17.16. 创建新的 **Vnet**

你可以使用 **xm** 来创建一个新的 Vnet：

```
xm vnet-create [configfile]
```

你必须指定一个配置文件来创建新的 Vnet。

17.17. 终止 **Vnet**

你也可以使用 **xm** 来终止现有的 Vnet：

```
xm vnet-delete [VnetID]
```

这个命令终止了你指定的 Vnet。

17.18. 创建域安全性标签（**Domain Security Label**）

你可以使用 **xm** 来创建一个域安全性标签：

```
xm addlabel [labelname] [domain-id] [configfile]
```

17.19. 测试域资源

你可以使用 **xm** 来测试某个域是否可以访问它自己的资源:

```
xm dry-run [configfile]
```

这个命令检查你的配置文件里列出的每个资源。它列出了每个资源的状态和最后的 security decision。

17.20. 显示系统资源

你可以使用 **xm** 来查看系统资源:

```
xm resources
```

输出显示了系统里的域的资源。

17.21. 配置 Credit Scheduling

你可以用 **xm** 来配置信用调度程序（credit scheduler）的参数:

```
xm sched-credit -d <domain> [ -w [=WEIGHT] | -c [CAP] ]
```

你可以用[-w] 选项配置权重（Weight）。你可以用 [-c] 选项配置峰值（Cap）。

17.22. 创建一个新的虚拟块设备

你可以使用 **xm** 来创建一个新的虚拟块设备：

```
xm block-attach [domain-id] [bedomain-id] [fe-dev] [be-dev] [mode]
```

即使客户机在运行的时候，你都可以附加（或分离）虚拟设备。下面是五个参数选项：

参数	描述
[domain-id]	附加到设备的客户机域的 domain-id。
[be-dev]	被导出的后台域里的设备
[fe-dev]	呈现给客户机域的设备
[mode]	客户机域的设备访问模式
[bedomain-id]	作为设备宿主的后台域

表 **17.5.** 新块设备参数

17.23. 终止虚拟块设备

你可以用 **xm** 来销毁一个现有的虚拟块设备：

```
xm block-detach [domain-id] [DevID]
```

这个命令销毁了你指定的虚拟块设备。

17.24. 安全性

17.24.1. 删除域安全性标签

你可以用 xm 来删除一个域安全性标签：

```
xm rmlabel [domain-id] [configfile]
```

这个命令删除了配置文件里的 acm_policy 标签条目。

17.24.2. 创建资源安全性标签

你可以用 **xm** 来创建一个资源安全性标签:

```
xm addlabel [labelname] res [resource] [policy]
```

17.24.3. 删除资源安全性标签

你可以用 **xm** 来删除一个资源安全性标签:

```
mx rmlabel [domain-id] res [resource]
```

这个命令删除了全局资源文件。

17.24.4. 配置访问控制

红帽虚拟化系统的访问控制由两个主要组件组成。Access Control Policy (ACP) 定义访问规则和安全性标签。当域请求访问资源时，Access Control Module (ACM) 应用这个策略并做出访问控制的决定。ACM 依照域安全性标签决定访问权限。然后，ACP 启用安全性标签和访问规则并把它们分配给域和资源。ACP 使用两个不同的标签管理方式:

标签	描述
Simple Type Enforcement	ACP 翻译这个标签并把访问权限分配给请求虚拟或物理访问的域。安全策略控制着域之间的访问并把正确的标签分配给不同的域。在缺省情况下，Simple Type Enforcement 方式是禁止的。
Chinese Wall	Chinese Wall 安全策略控制和响应域的访问请求。

表 **17.6. ACP** 标签管理

策略（policy）是一个本地路径和策略 XML 文件（相对于全局策略根目录）的列表。例如，域文件 chinese_wall.client_V1 从属于策略文件 /example/chinese_wall.client_v1.xml。

红帽虚拟化包括以下这些参数，这允许你管理安全策略和给域分配标签：

17.24.5. 创建策略

你可以用 **xm** 来创建一个二进制策略：

```
xm makepolicy [policy]
```

这个命令创建了二进制策略并保存为二进制文件 [policy.bin]。

17.24.6. 加载策略

你可以使用 **xm** 来加载一个二进制策略：

```
xm loadpolicy [policybin]
```

17.24.7. 为引导配置文件创建策略

你可以使用 **xm** 来建立一个二进制策略并把它加入到引导配置文件里：

```
xm cfgbootpolicy [kernelversion]
```

这个命令把二进制策略复制到 /boot 目录并修改 /boot/grub/menu.1st 文件里相对应的行。

17.24.8. 创建标签

你可以使用 **xm** 来创建一个标签：

```
xm addlabel [configfile] [policy]
```

把一个安全性标签加入到域配置文件里。它也校验不同的策略定义是否与对应的标签名相匹配。

17.24.9. 显示策略标签

你可以使用 xm 来查看策略标签：

```
xm labels [policy] [type=dom | res | any]
```

这个命令显示了你在创建策略时指定的类型的标签（缺省是 dom）。

17.24.10. 显示域安全性标签

你可以使用 xm 来查看域的安全性标签：

```
xm getlabel domain-id [configfile]
```

17.24.11. 显示资源安全性标签

你可以用 xm 来查看资源的安全性标签：

```
xm getlabel res [resource]
```

17.24.12. 配置访问控制的安全性

要启用红帽虚拟化访问安全性，你必须修改 xen_source__dir/Config.mk 里的这些参数：

```
ACM_SECURITY ?= y
```

```
ACM_DEFAULT_SECURITY_POLICY ? =

ACM_CHINESE_WALL__AND_SIMPLE_TYPE_ENFORCEMENT_POLICY
```

17.24.13. 编译安全性策略

这个例子演示了怎样成功地编译一个安全性策略：

```
xm makepolicy chinesewall_ste.client_v1
```

这个命令创建了 /etc/xen/acm-security/policies/example/chinesewall_ste 目录下的 client_v1.map 和 client_v1.bin 文件。

17.24.14. 加载安全性策略

你可以使用 **xm** 来激活 client_v1.bin ：

```
xm loadpolicy example.chwall_ste.client_v1
```

17.24.15. 配置引导安全性策略

你可以使用 **xm** 配置引导装载程序来加载 client_v1.bin ：

```
xm cfgbootpolicy chinesewall_ste.client_v1
```

这个命令使 ACM 使用这个标签来引导红帽虚拟化系统。

17.24.16. 显示安全性标签

你可以使用 xm 来查看定义的标签：

```
xm labels  chinesewall_ste.client_v1 type=dom
```

下面的输出显示了所有 dom 的策略：

```
dom_StorageDomain

dom_SystemManagement

dom_NetworkDomain

dom_QandA

dom_R&D
```

17.24.17. 附加安全性标签

你可以用 **xm** 来把安全性标签附加到域配置文件（这个示例使用了 SoftwareDev 标签）：

```
xm addlabel myconfig.xm dom_SoftwareDev
```

附加安全性标签确保了域不会与其他 non-SoftwareDev 用户域共享数据。这个示例包括了 myconfig.xm 配置文件，它代表了一个运行与 SoftwareDev 的基础架构相关的任务的域。

编辑相应的配置文件并校验 addlabel 命令是否正确地把 access_control 条目（以及相关的参数）添加到了这个文件的最后：

```
kernel = "/boot/vmlinuz - 2.6 5 -xen"

ramdisk="/boot/U1_SoftwareDev_ramdisk img"

memory = 64

name = "SoftwareDev"

vif = [ '' ]

dhcp = "dhcp"

access_control = [policy=example.chwall_ste.client_v1, label=dom_SoftwareDev]
```

如果输出不正确，请进行相应的修改并保存这个文件。

第 18 章 用虚拟机管理者（Virtual Machine Manager）管理虚拟机

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

18.1. 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。

18.2. Open Connection 窗口

首先，这个窗口将出现并提示用户选择监控程序会话。非专有（non-privileged）用户可以初始化一个只读会话。根用户可以启动具有完全读写权限的会话。对于普通的应用，可以选择 `Local Xen host` 选项。你可以通过选择 `Other hypervisor` 并在 URL 字段键入 test:///default 来启动虚拟机管理者测试模式。在测试模式下，你可以连接至一个 libvirt dummy 监控程序。注意，虽然 `Remote Xen host screen` 是可见的，但是连接至这样的主机在 RHEL 5.0 里还没有实施。

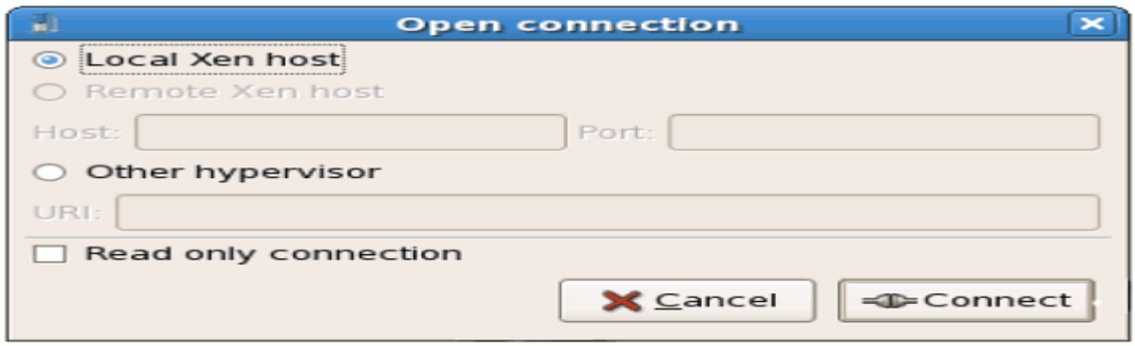

图 **18.1.** 虚拟机管理者 **Connection** 窗口

18.3. 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。

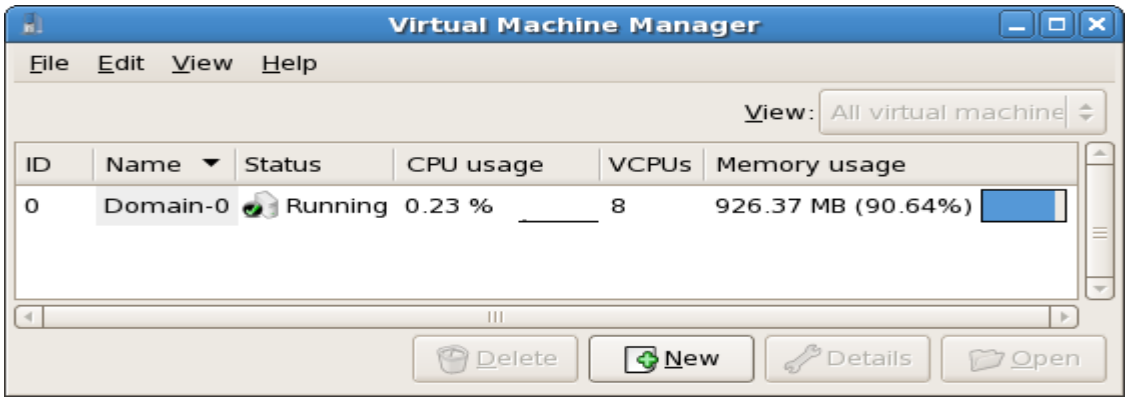

图 **18.2.** 虚拟机管理者主窗口

18.4. 虚拟机 Details 窗口

这个窗口显示红帽虚拟化系统里的客户机资源利用的图例和统计。UUID 字段显示了虚拟机的唯一全局标识符。

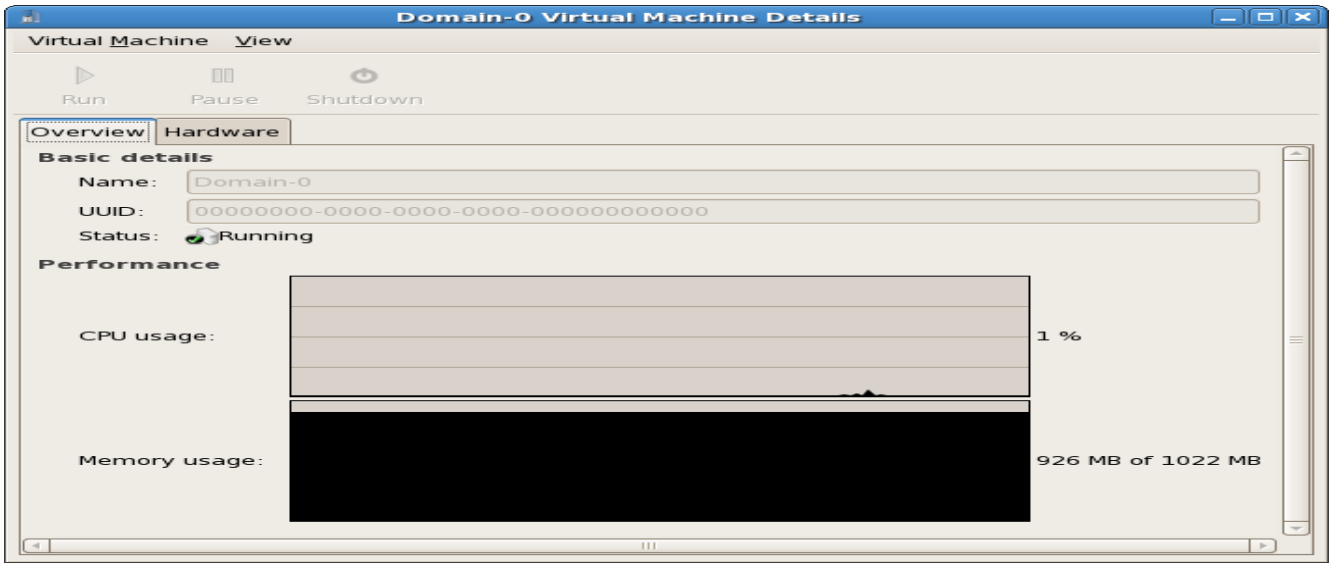

图 **18.3.** 虚拟机管理者 **Details** 窗口

18.5. 虚拟机图形化控制台

这个窗口显示了虚拟机的图形化控制台。并行和完全虚拟机使用不同的技术来导出它们的本地虚拟化帧缓冲，但是它们都使用 VNC 来为虚拟机管理者的控制台窗口可用。如果你的虚拟机被设置为需要验证，在显示前虚拟机图形化控制台会提示你输入密码。

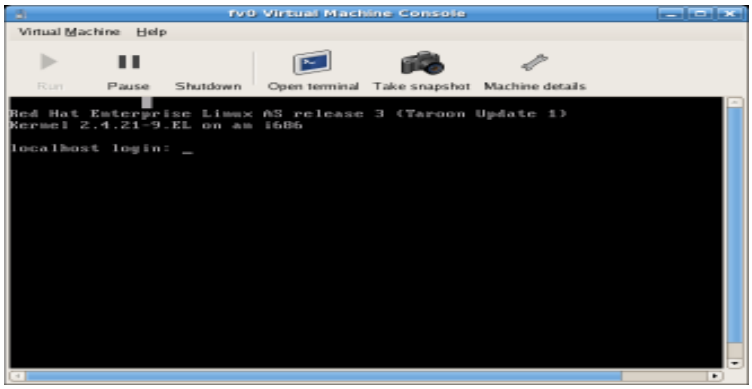

图 **18.4.** 图形化控制台

你的本地桌面可以阻止组合键（如 Ctrl+Alt+F11）被发送到客户机。你可以使用虚拟机管理者的 'sticky key' 能力来发送这些键序列。你必须按任何 modifier 键（如 Ctrl 或 Alt）三次，这个键才会被激活，直到下一个 non-modifier 键被按住。例如，你可以按键序列 'Ctrl Ctrl Ctrl Alt+F1' 来把 Ctrl-Alt-F11 发送给客户机。

18.6. 启动虚拟机管理者

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。

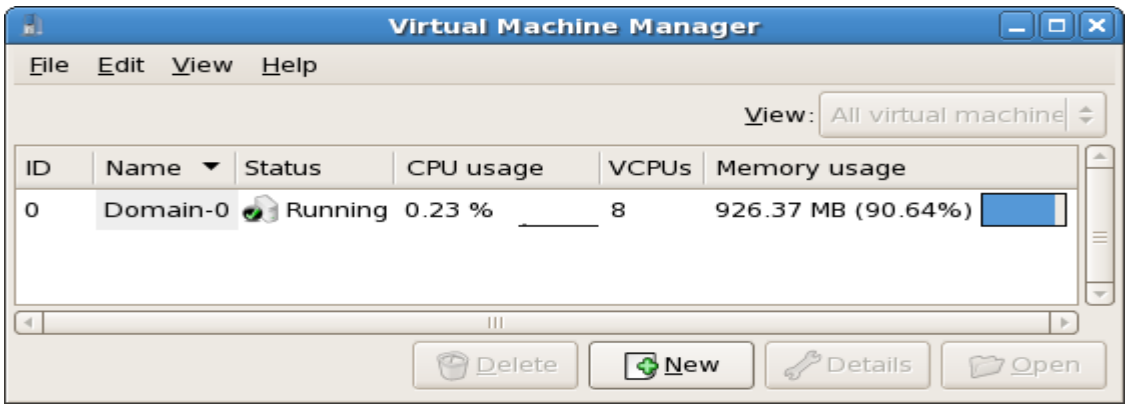

图 **18.5.** 启动虚拟机管理者

18.7. 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。

你可以使用红帽的虚拟机管理者来:

- 创建新的域。
- 配置或调整域的资源分配和虚拟硬件。
- 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:

你必须在需要虚拟化的所有系统里安装红帽企业 Linux 5.0、virt-manager 和内核软件包。所有系统都必须运行红帽虚拟化内核。

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 **18.1.** 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。

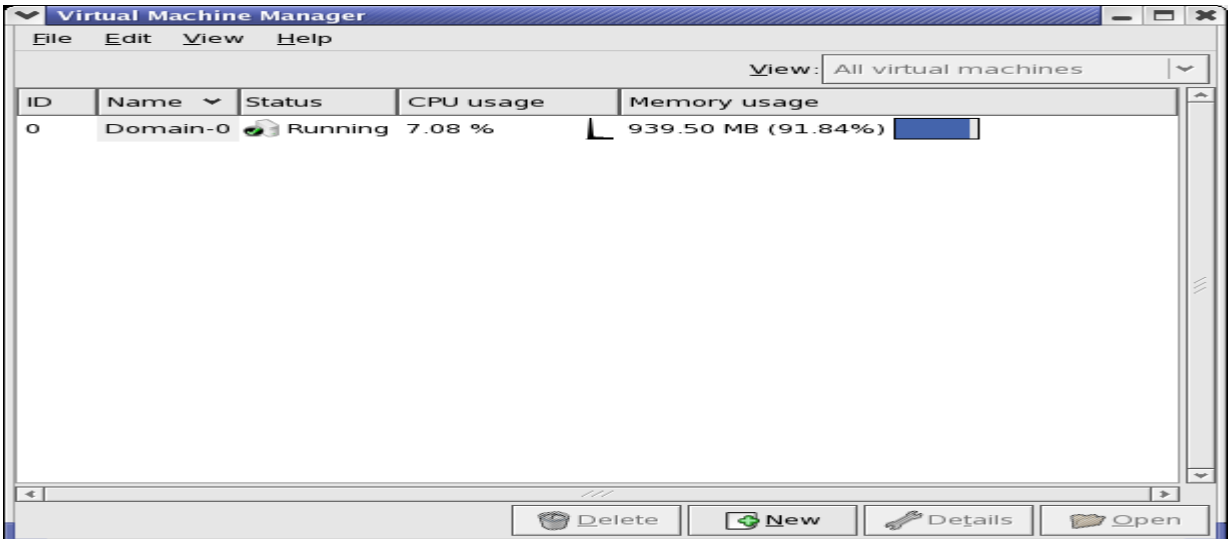

   图 **18.6.** 虚拟机管理者窗口

2. 在 **File** 菜单，选择 **New machine**。

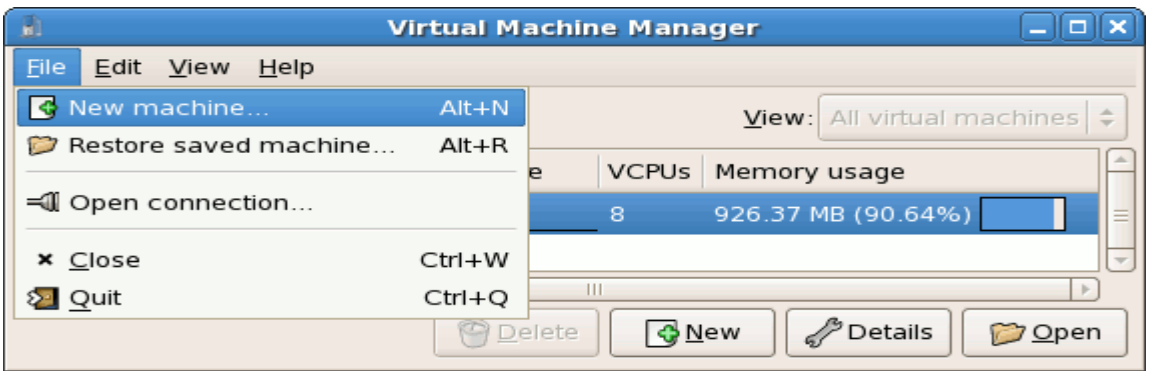

图 **18.7.** 选择一个新的机器

创建新虚拟机的向导将出现。

3. 点击 `Forward`。

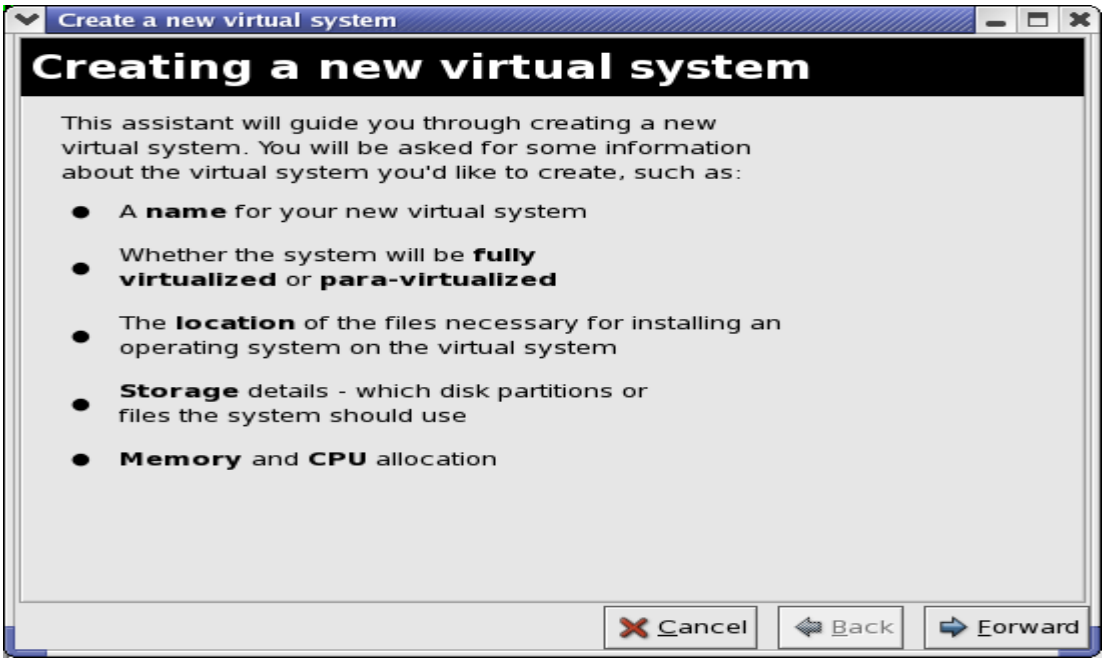

图 **18.8.** 创建新虚拟系统的向导

4. 输入新虚拟系统的名字并点击 `Forward`。

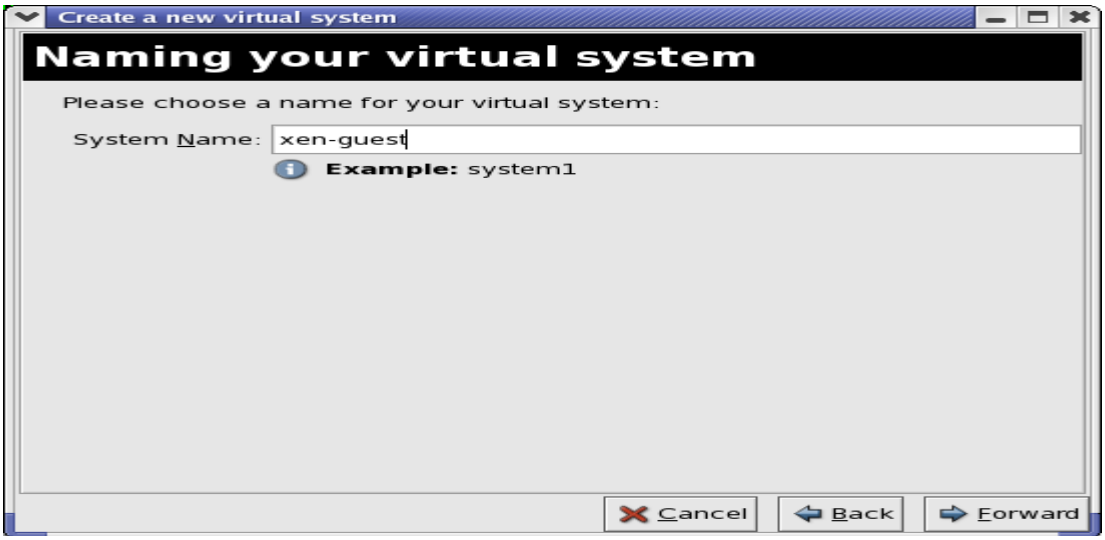

图 **18.9.** 命名虚拟系统

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` 。

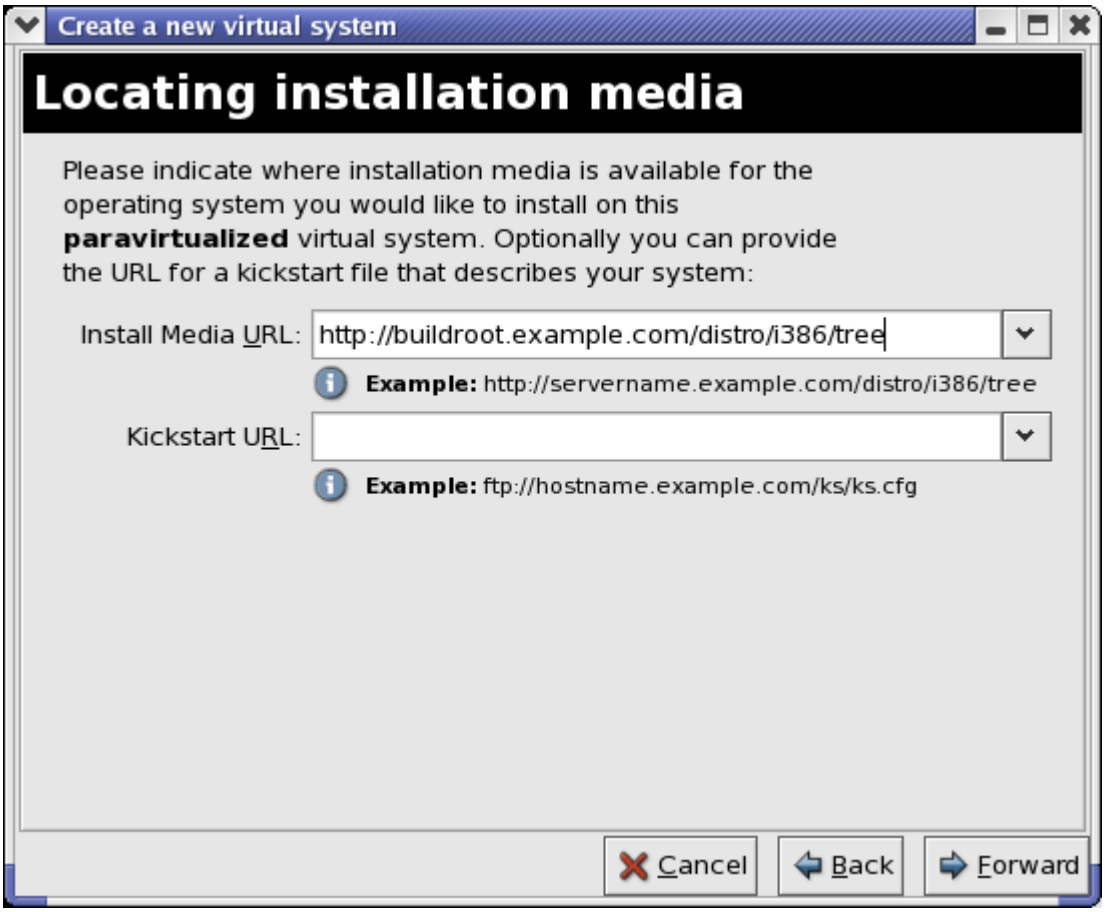

图 **18.10.** 定位安装介质

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。

缺省的 SELinux 策略允许把 xen 磁盘映像存放在 */var/lib/xen*。如果你启用了 SELinux 并想为虚拟磁盘指定其他自定义的路径，你需要相应地修改 SELinux 策略。

打开一个终端并创建 /xen 目录，然后用 restorecon -v /xen 命令设定 SELinux 策略。

指定虚拟磁盘的位置和大小，然后点击 `Forward`。

图 **18.11.** 分配存储空间

7. 选择分配给客户机的内存和虚拟 CPU 数量，然后点击 `Forward`。

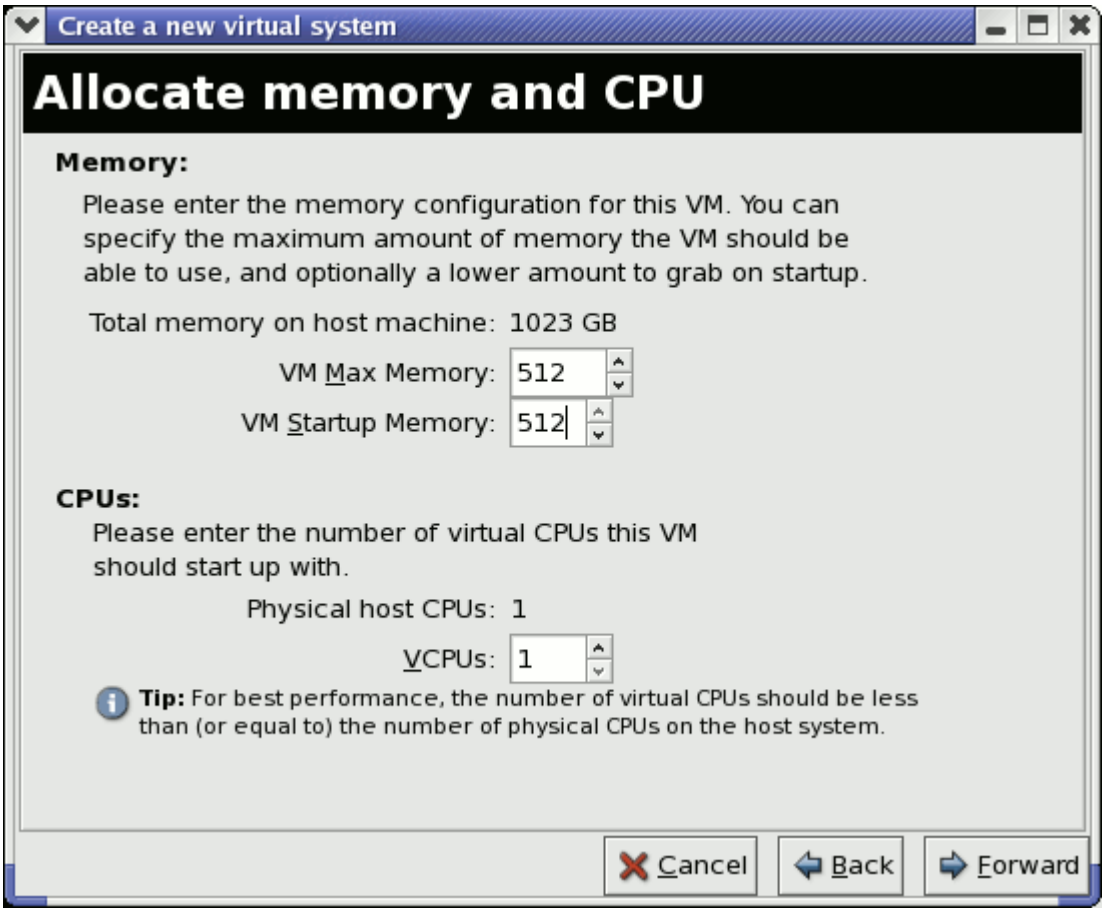

图 **18.12.** 分配内存和 **CPU**

8. 选择 Forward 来打开控制台和开始安装的文件。

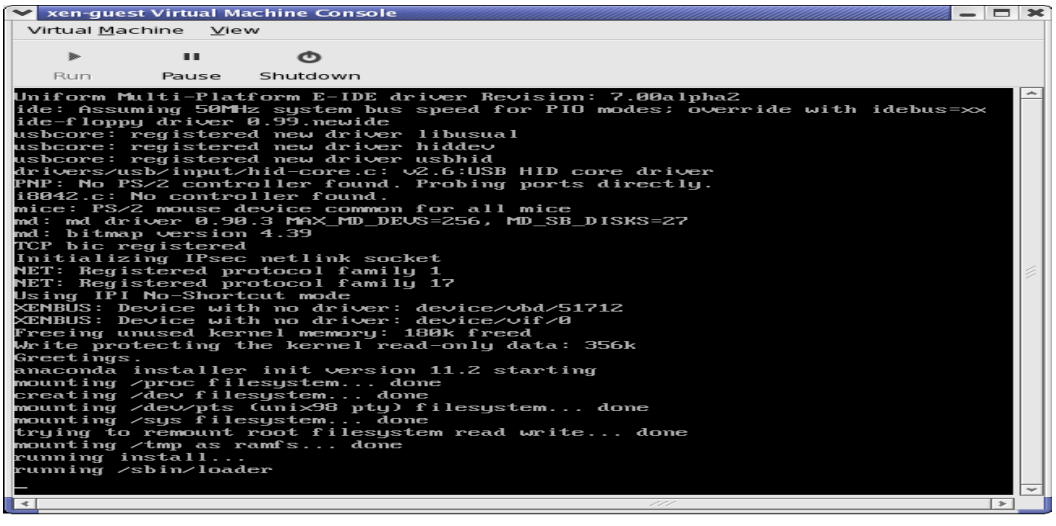

图 **18.13.** 分配内存和 **CPU**

9. 在这个窗口里完成安装。

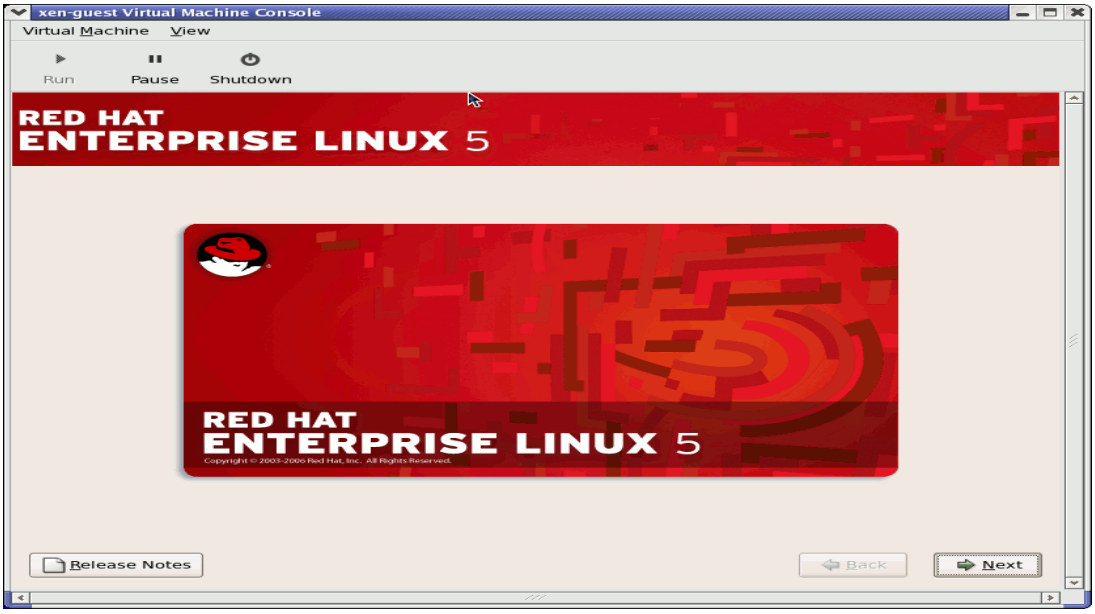

图 **18.14.** 安装开始...

10. 键入 xm create -c xen-guest 来启动红帽企业 Linux 5.0 客户机。右击虚拟机管理者里的客户机条目，选择 Open 来打开虚拟控制台。

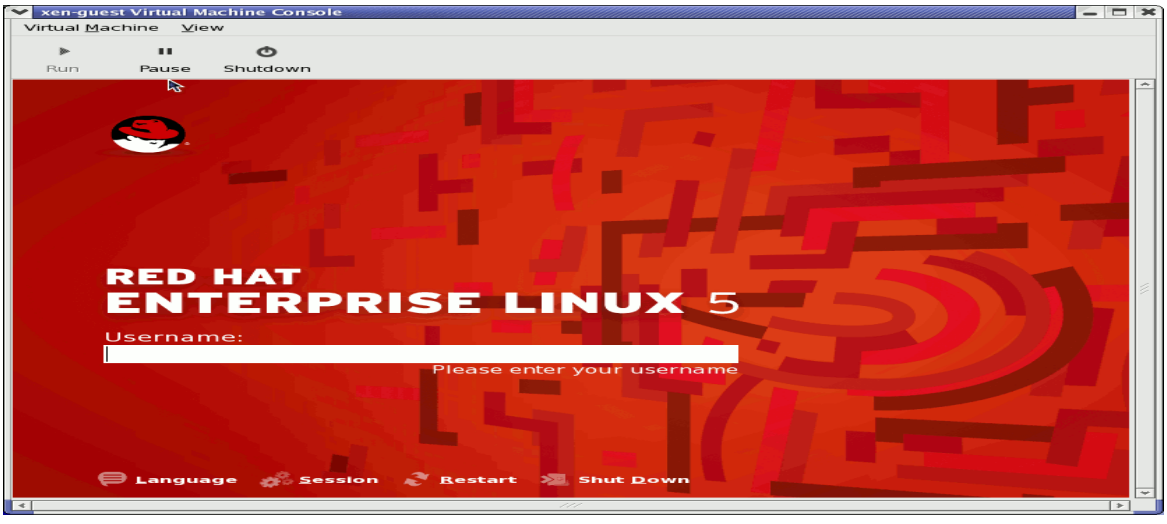

图 **18.15.** 红帽企业 **Linux 5.0**（客户机）

11. 输入用户名和密码来继续使用虚拟机管理者。

18.7. 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。

你可以使用红帽的虚拟机管理者来:

- 创建新的域。
- 配置或调整域的资源分配和虚拟硬件。
- 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:

你必须在需要虚拟化的所有系统里安装红帽企业 Linux 5.0、virt-manager 和内核软件包。所有系统都必须运行红帽虚拟化内核。

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 **18.1.** 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。

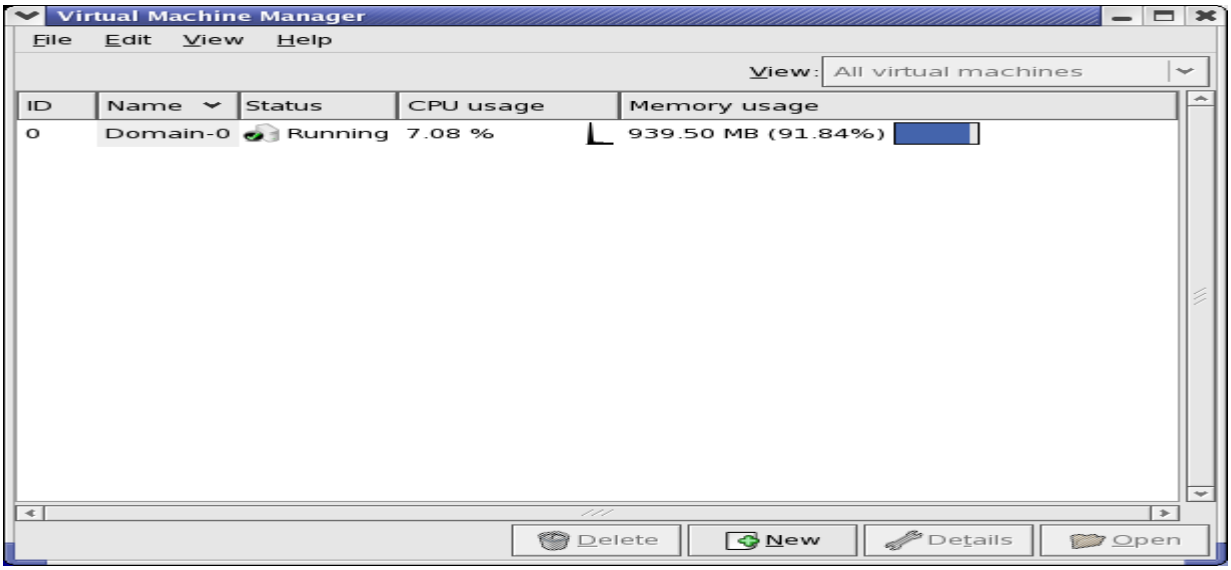

图 **18.6.** 虚拟机管理者窗口

2. 在 `File` 菜单，选择 `New machine`。

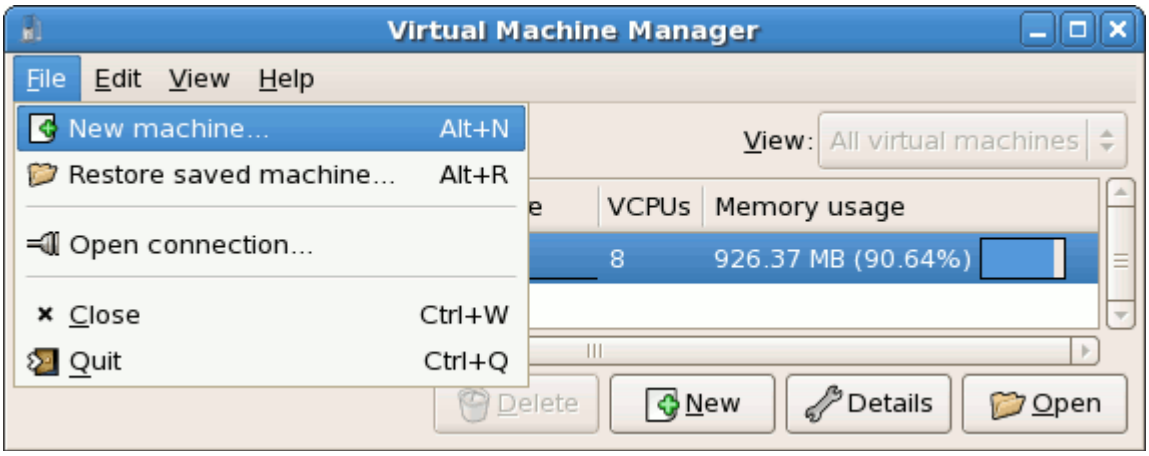

图 **18.7.** 选择一个新的机器

创建新虚拟机的向导将出现。

3. 点击 Forward。

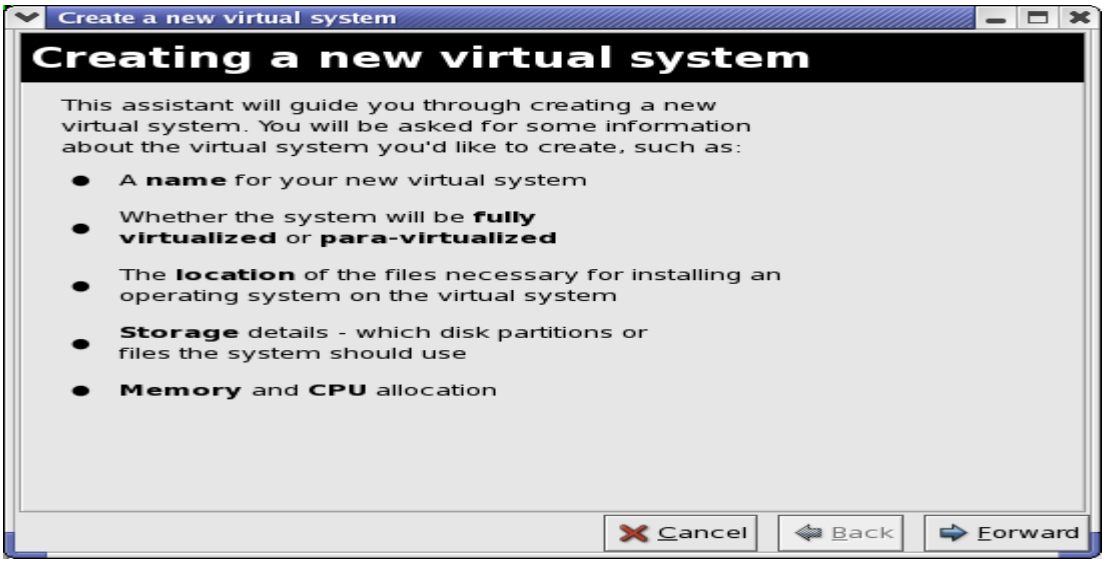

图 **18.8.** 创建新虚拟系统的向导

4. 输入新虚拟系统的名字并点击 Forward。

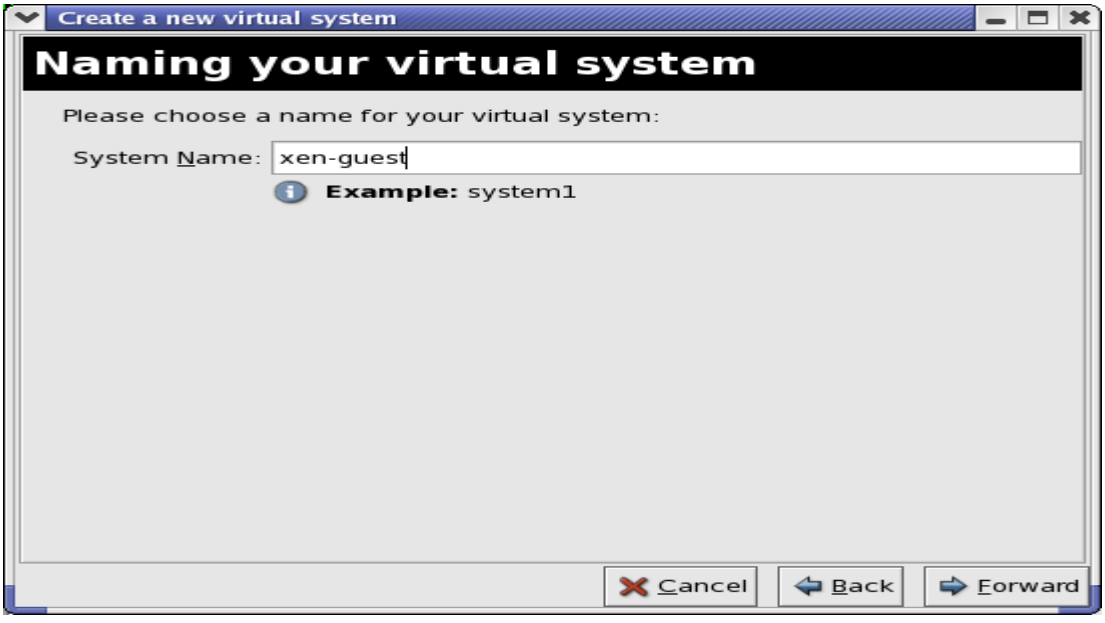

图 **18.9.** 命名虚拟系统

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 **18.10.** 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。

缺省的 SELinux 策略允许把 xen 磁盘映像存放在 */var/lib/xen*。如果你启用了 SELinux 并想为虚拟磁盘指定其他自定义的路径，你需要相应地修改 SELinux 策略。

打开一个终端并创建 /xen 目录，然后用 restorecon -v /xen 命令设定 SELinux 策略。

指定虚拟磁盘的位置和大小，然后点击 Forward。

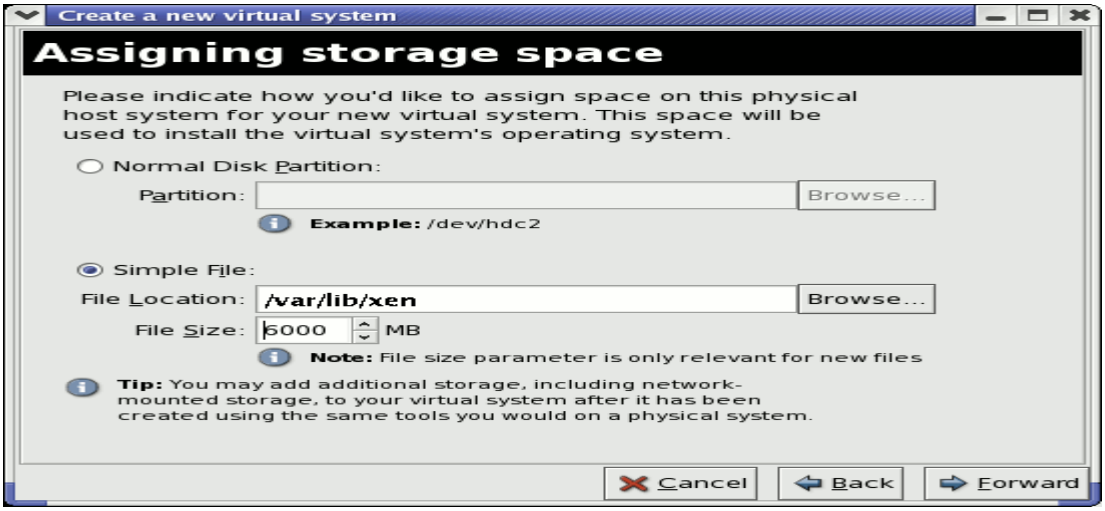

图 **18.11.** 分配存储空间

7. 选择分配给客户机的内存和虚拟 CPU 数量，然后点击 `Forward`。

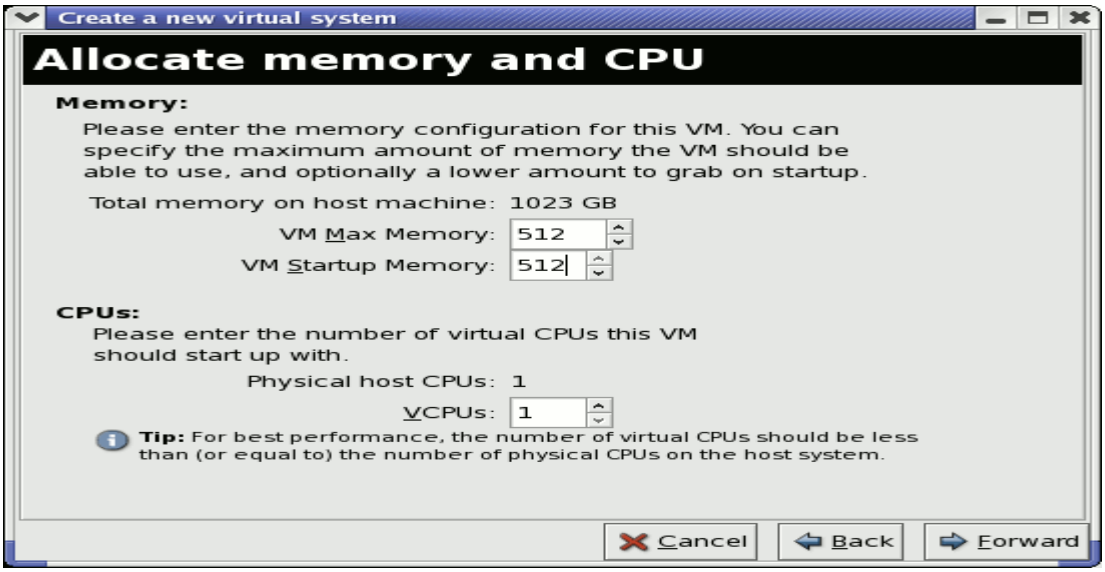

图 **18.12.** 分配内存和 **CPU**

8. 选择 `Forward` 来打开控制台和开始安装的文件。

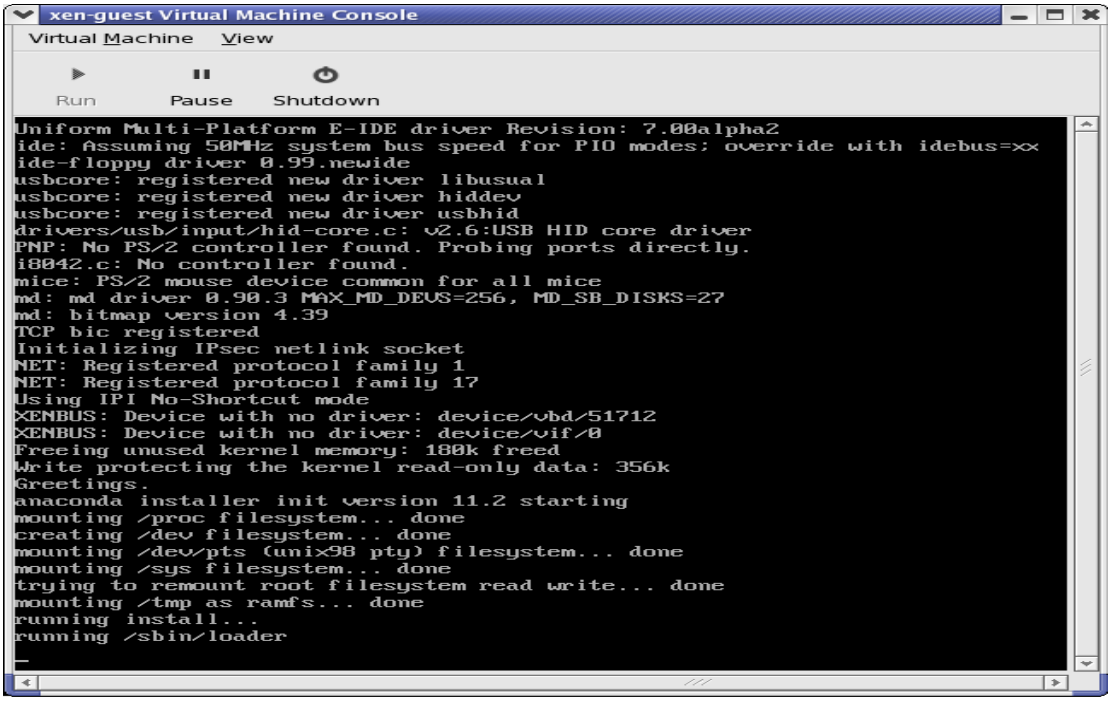

图 **18.13.** 分配内存和 **CPU**

9. 在这个窗口里完成安装。

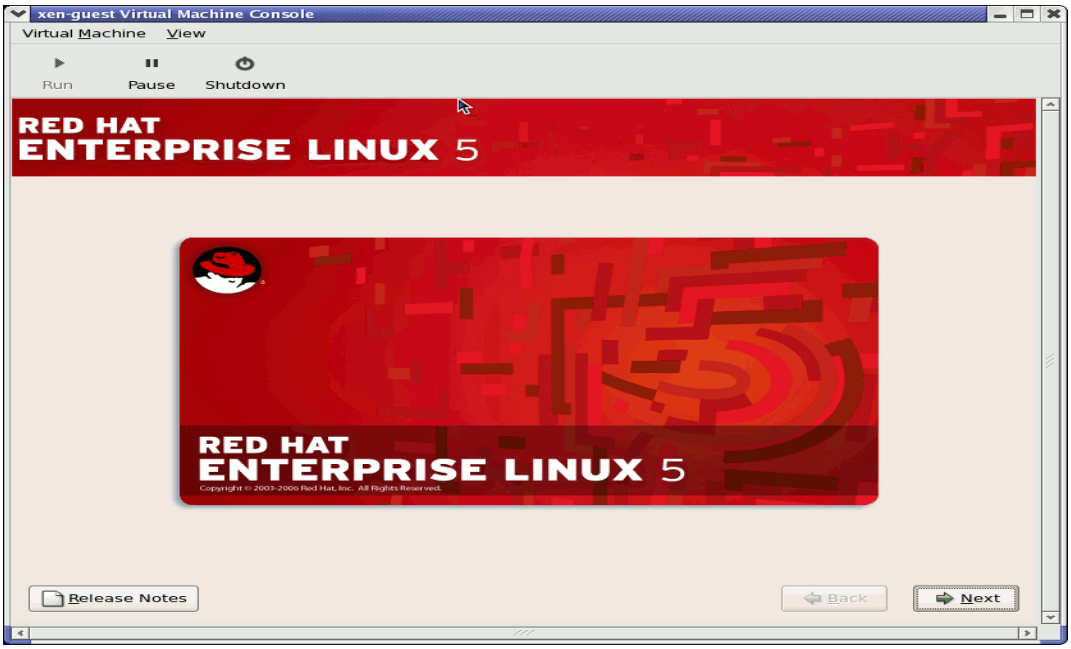

图 **18.14.** 安装开始...

10. 键入 xm create -c xen-guest 来启动红帽企业 Linux 5.0 客户机。右击虚拟机管理者里的客户机条目，选择 Open 来打开虚拟控制台。

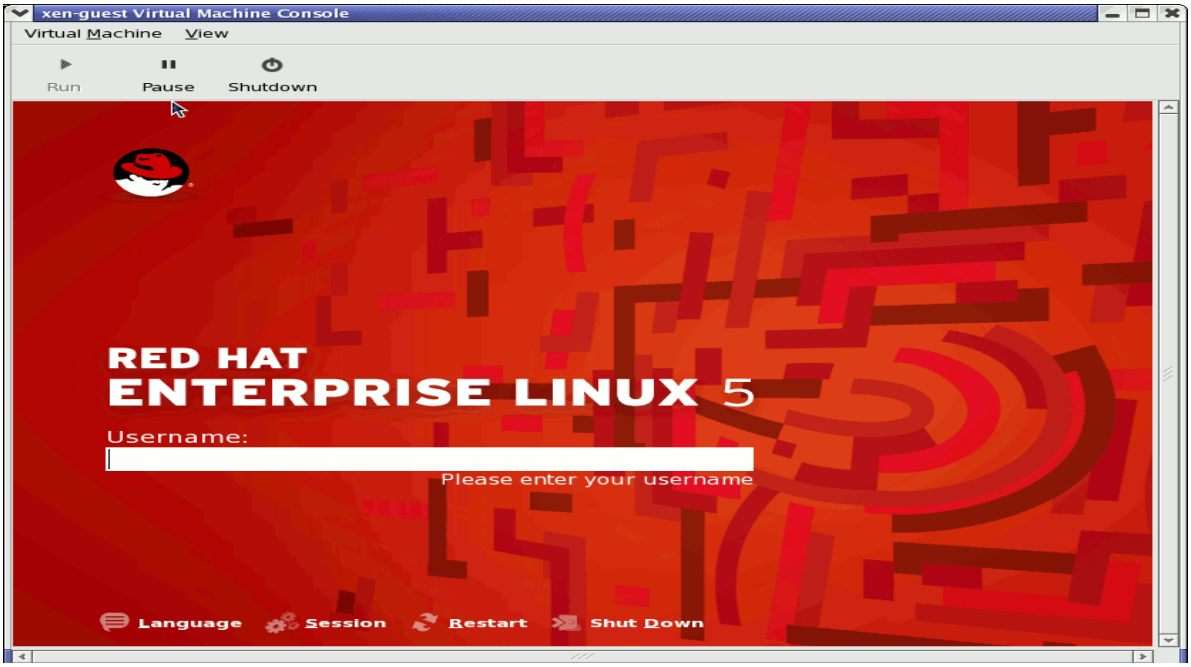

图 **18.15.** 红帽企业 **Linux 5.0**（客户机）

11. 输入用户名和密码来继续使用虚拟机管理者。

18.8. 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。

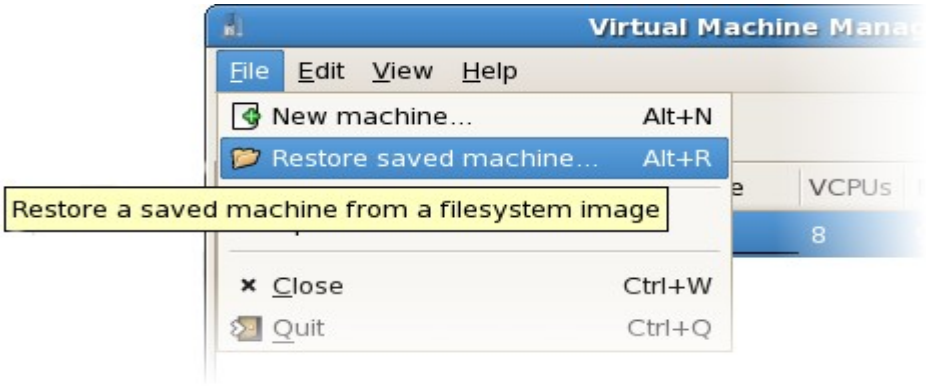

图 **18.16.** 恢复虚拟机

2. 恢复虚拟机主窗口将出现。

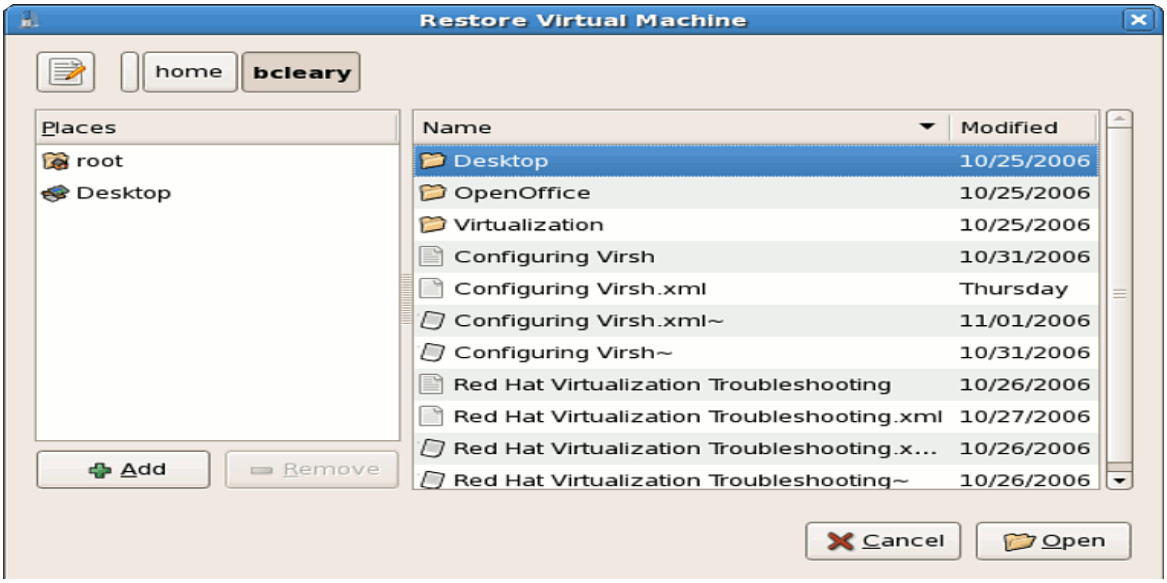

图 **18.17.** 选择保存的虚拟机会话

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。

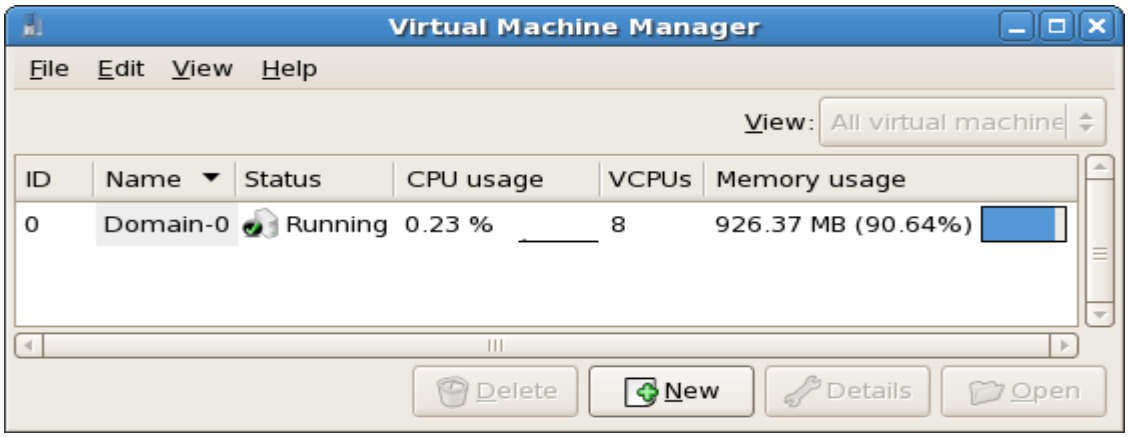

图 **18.18.** 被恢复的虚拟机管理者会话

18.9. 显示虚拟机的细节

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。

查看虚拟系统的细节：

1. 在虚拟机管理者主窗口，高亮显示你要查看的虚拟机。

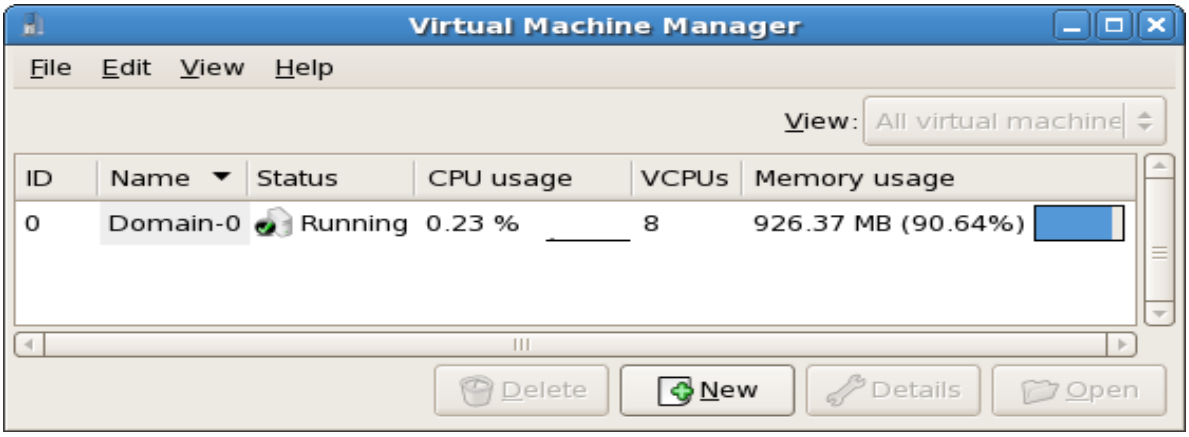

图 **18.19.** 选择要显示的虚拟机

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）。

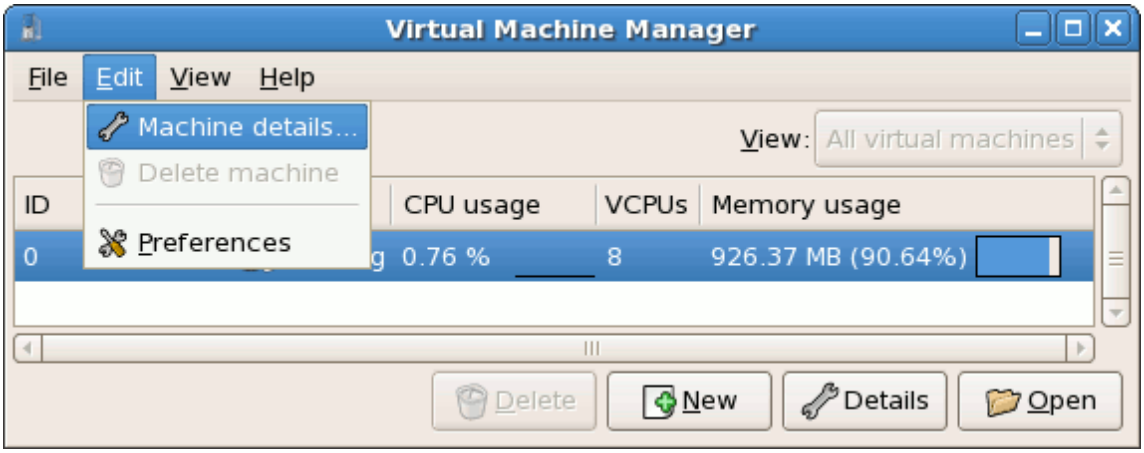

图 **18.20.** 显示虚拟机细节菜单

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。

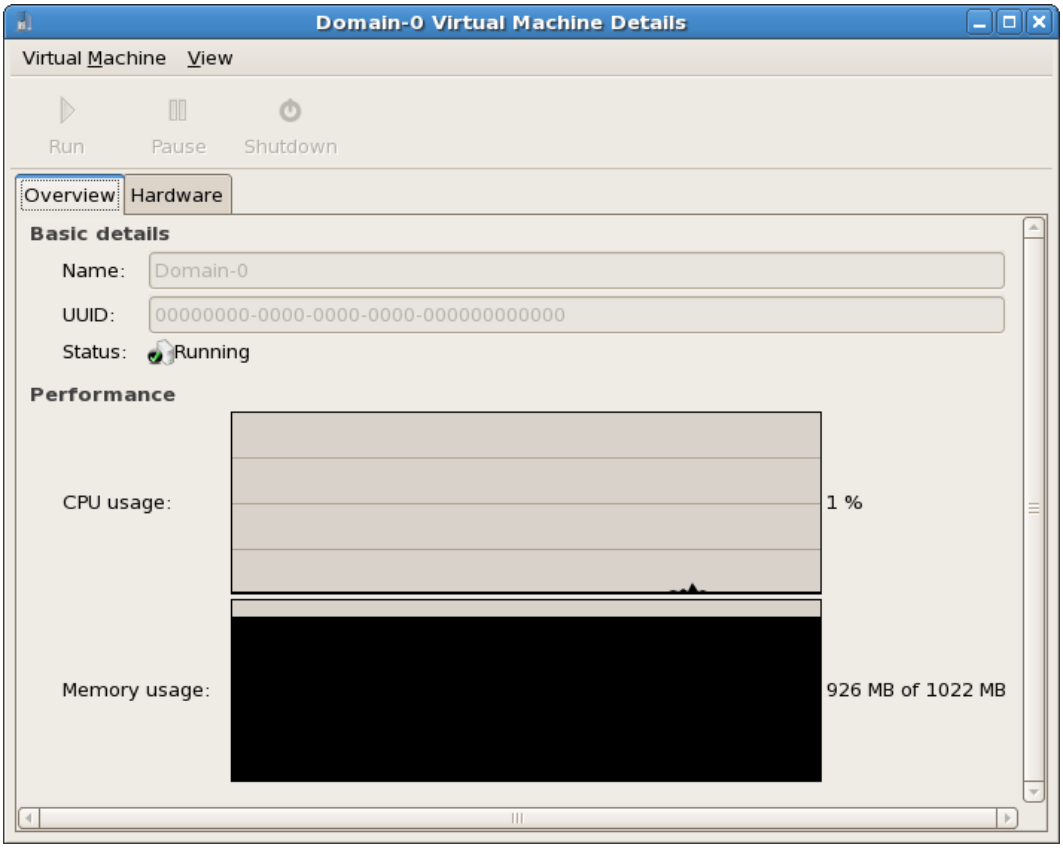

图 **18.21.** 显示 **Virtual Machine Details Overview**

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

Virtual Machine Details Hardware 窗口将出现。

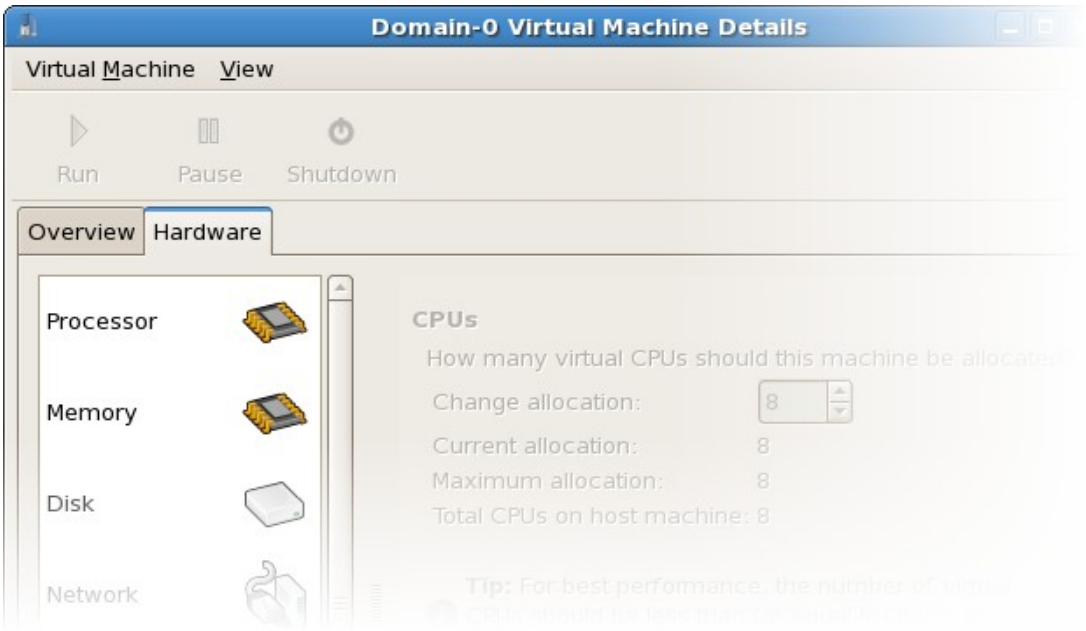

图 **18.22.** 显示 **Virtual Machine Details Hardware**

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 **18.23.** 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。

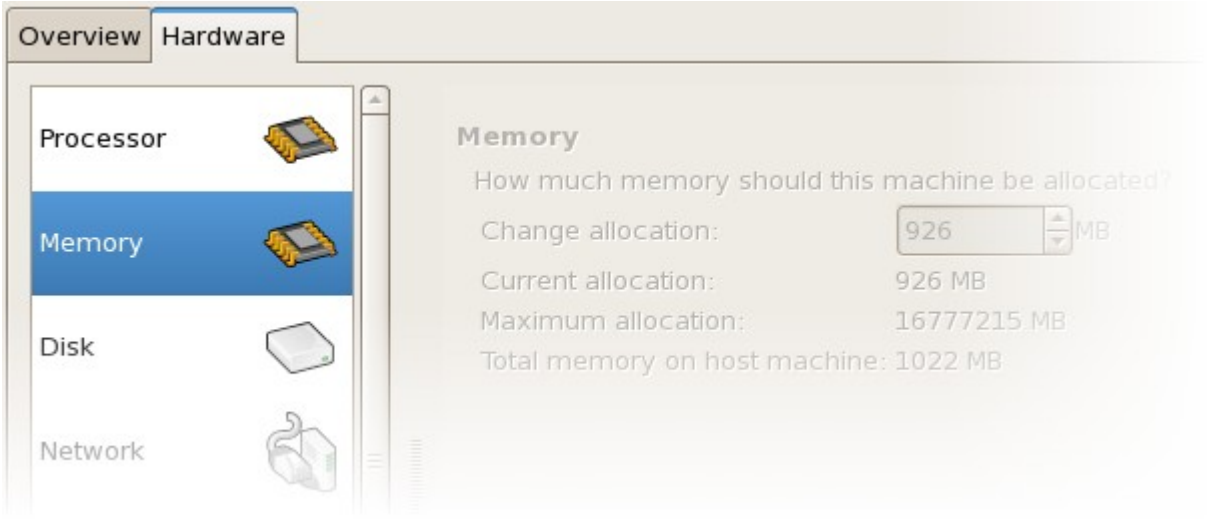

图 **18.24.** 显示内存分配情况

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

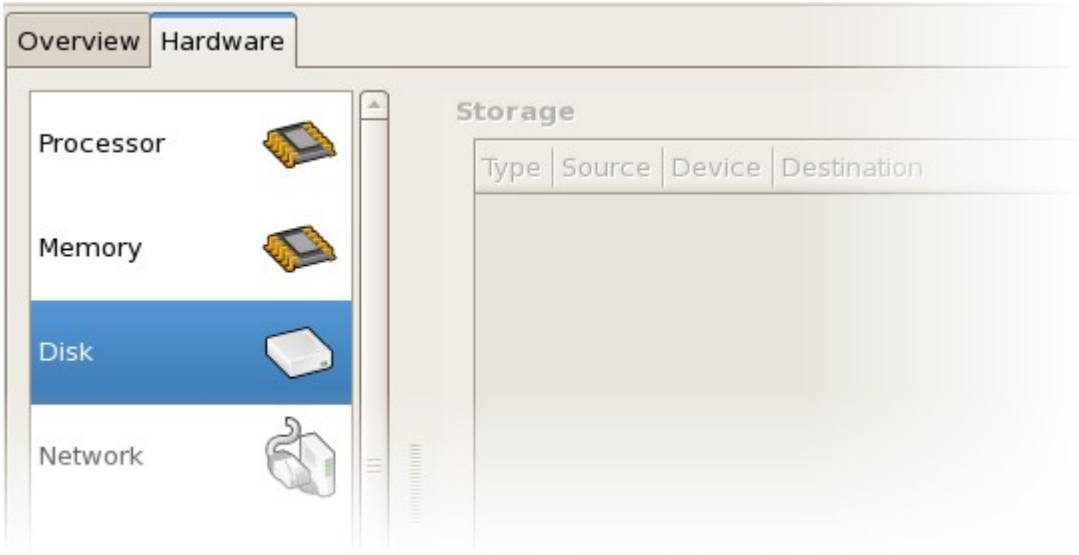

图 **18.25.** 显示磁盘配置

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

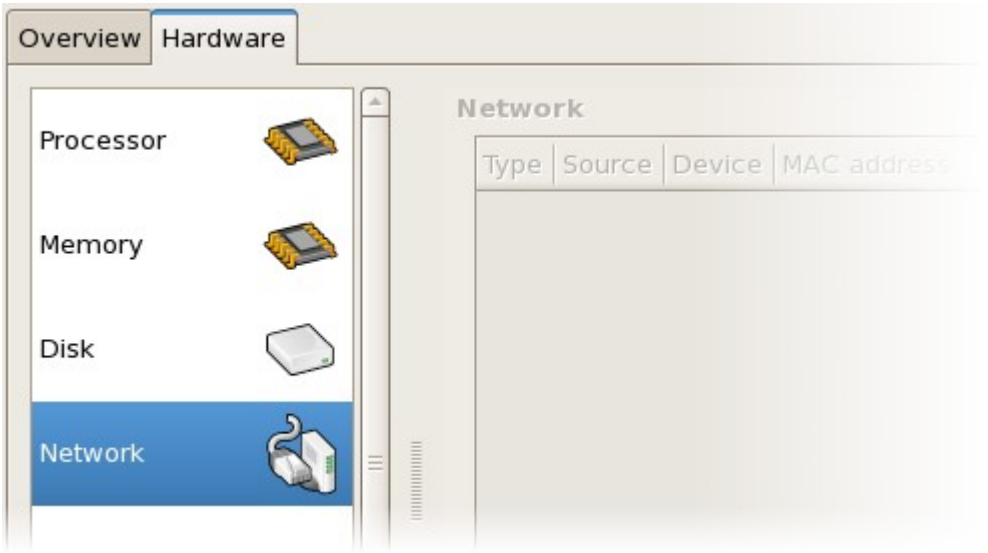

图 **18.26.** 显示网络配置

18.10. 配置状态监控

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`。

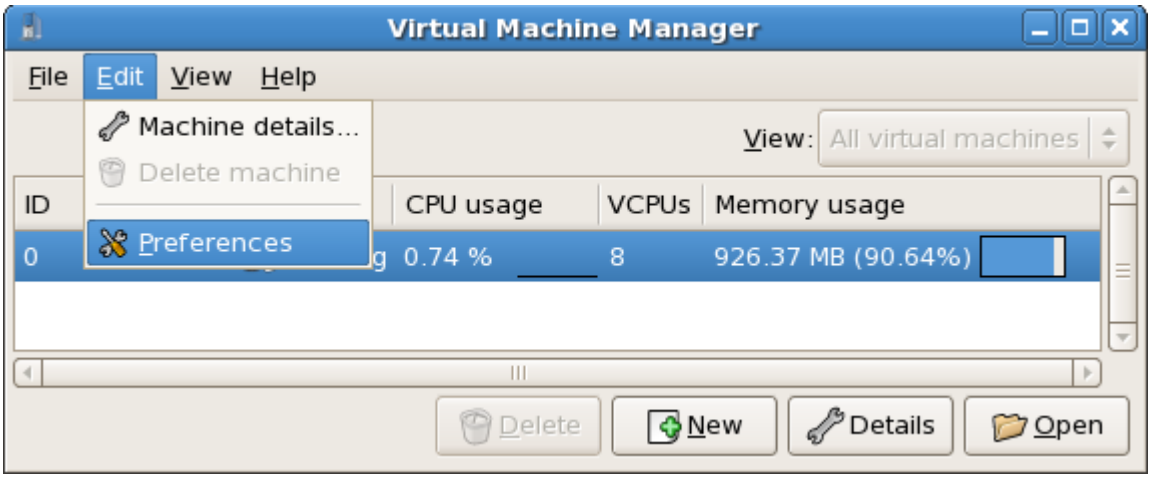

图 **18.27.** 改变虚拟机的首选项

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）。

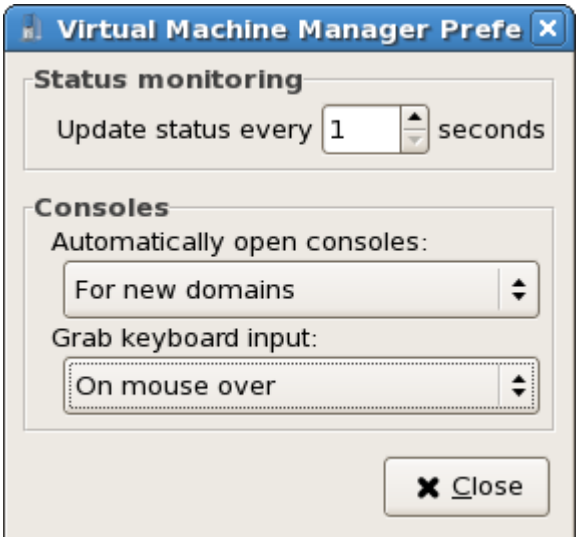

图 **18.28.** 配置状态监控

3. 在控制台区，指定怎样打开控制台和输入设备。

18.11. 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。

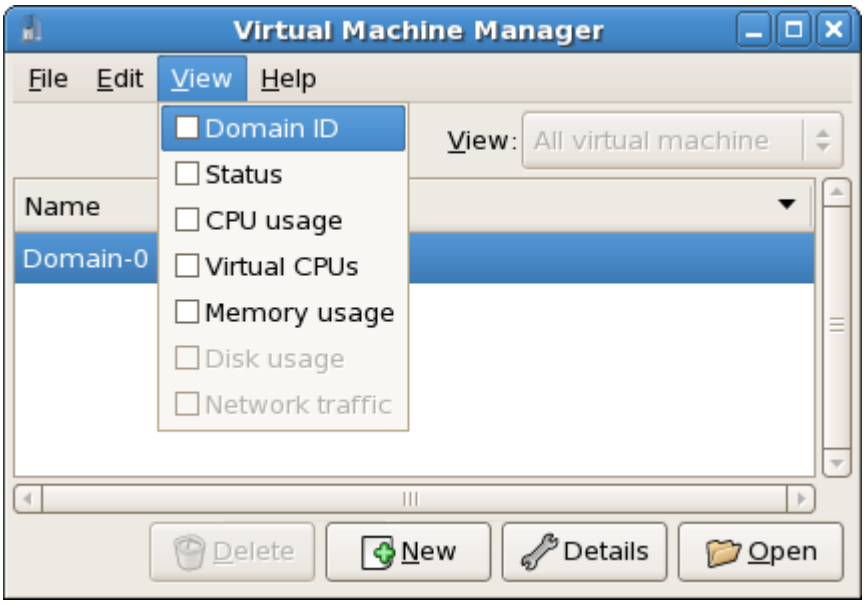

图 **18.29.** 显示 **Domain-ID**

2. 虚拟机管理者列出了系统里所有域的 Domain ID。

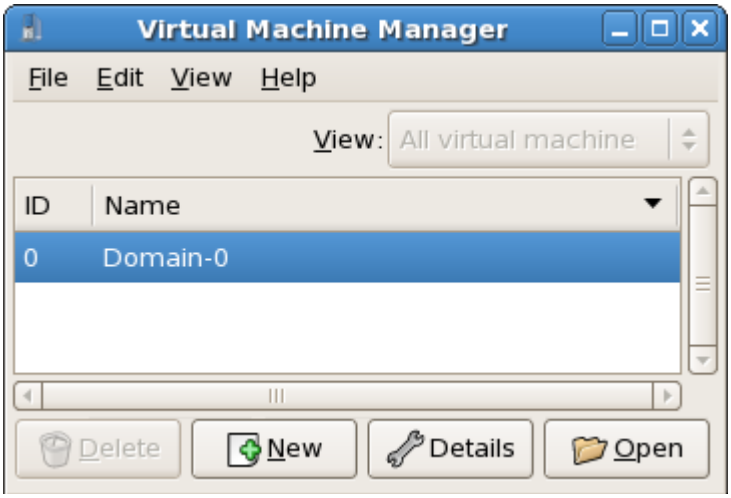

图 **18.30.** 显示 **Domain-ID**

18.12. 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。

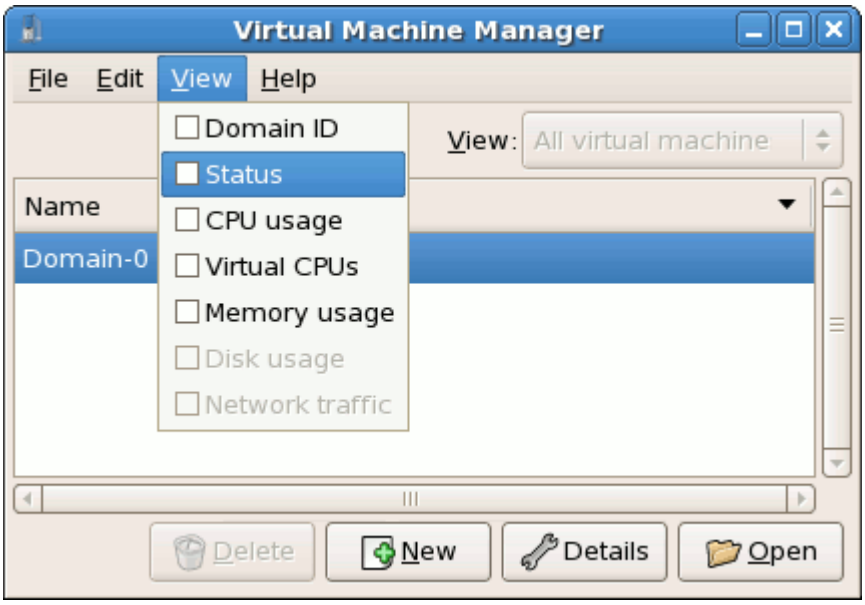

图 **18.31.** 显示虚拟机状态

2. 虚拟机管理者列出了系统里所有虚拟机的状态。

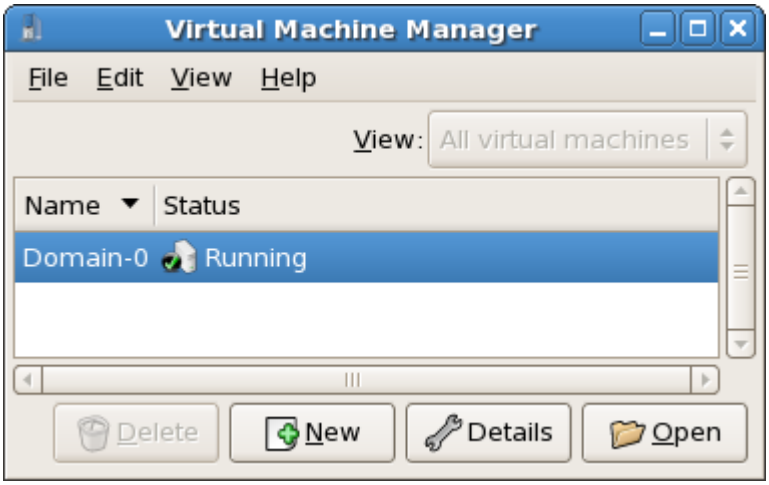

图 **18.32.** 显示虚拟机状态

18.13. 显示虚拟 CPU

查看系统里所有虚拟机的虚拟 CPU 的数量：

1. 在 `View` 菜单，选定 `Virtual CPUs` 复选框。

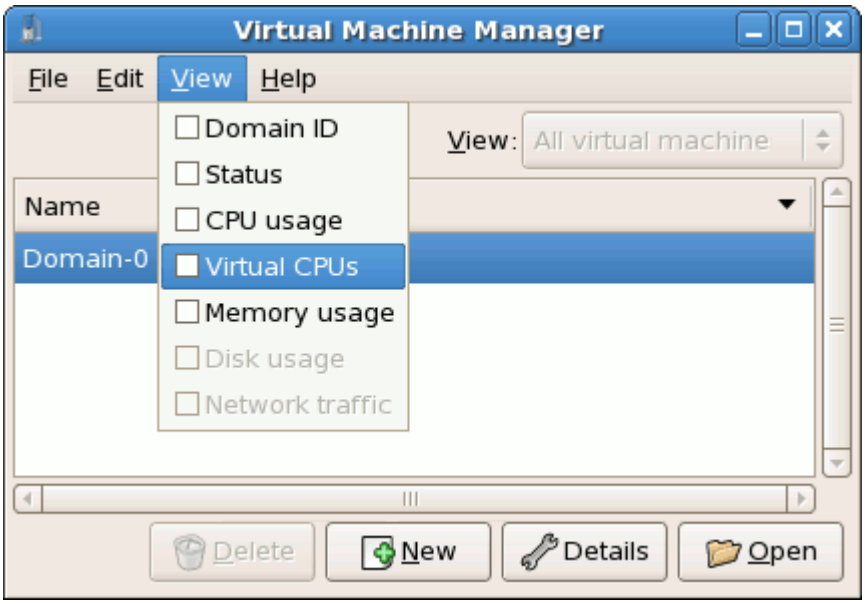

图 **18.33.** 显示虚拟 **CPU**

2. 虚拟机管理者将列出系统里所有虚拟机的虚拟 CPU。

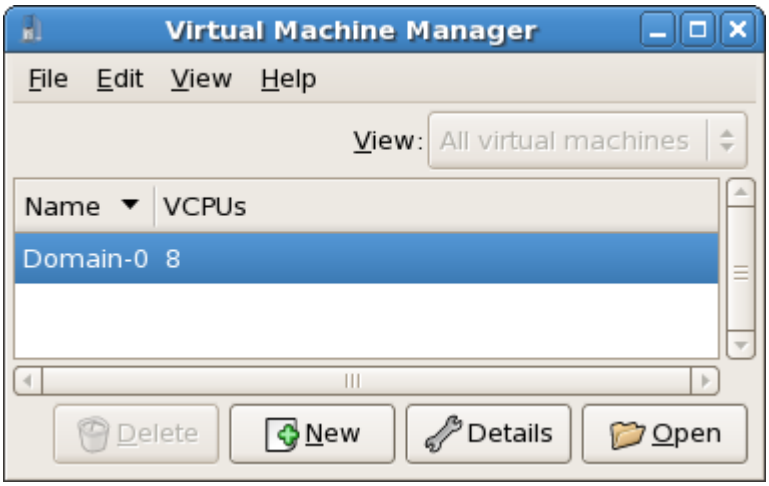

图 **18.34.** 显示虚拟 **CPU**

18.14. 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:

1. 在 `View` 菜单里，选定 `CPU Usage` 复选框。

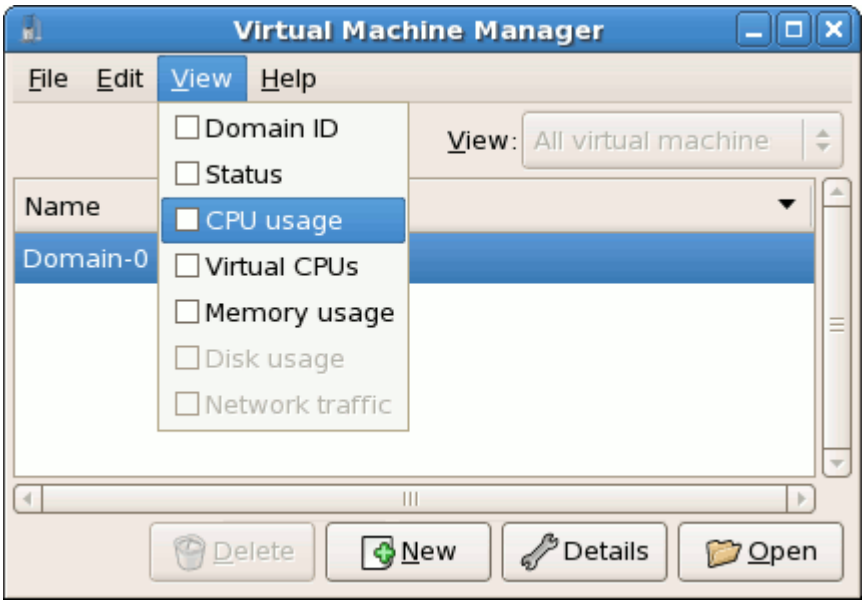

图 **18.35.** 显示 **CPU** 的使用情况

2. 虚拟机管理者将列出系统里所有虚拟机的 CPU 使用百分比。

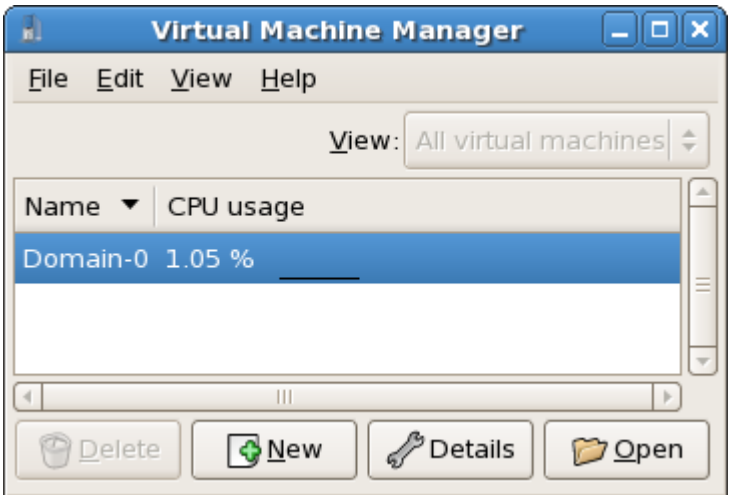

图 **18.36.** 显示 **CPU** 的使用情况

18.15. 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：

1. 在 `View` 菜单里，选定 `Memory Usage`复选框。

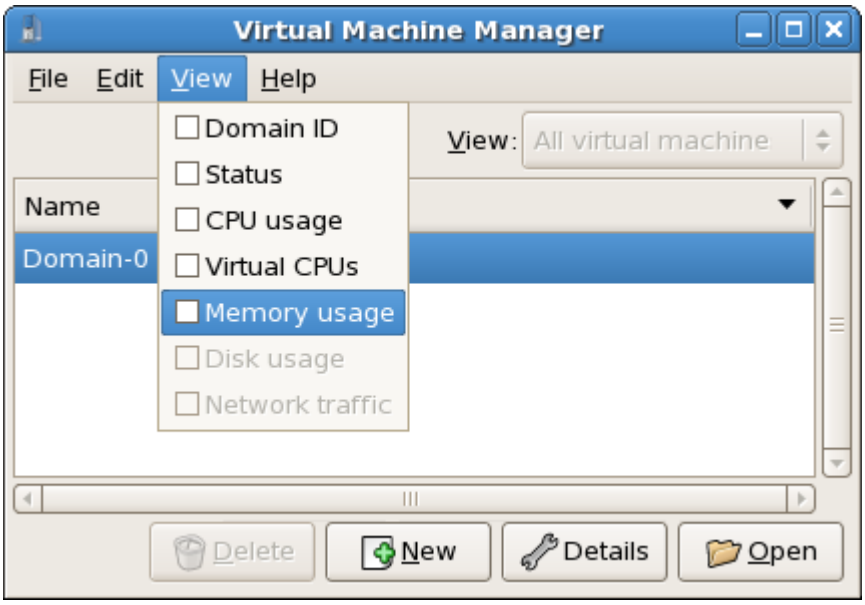

图 **18.37.** 显示内存使用情况

2. 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。

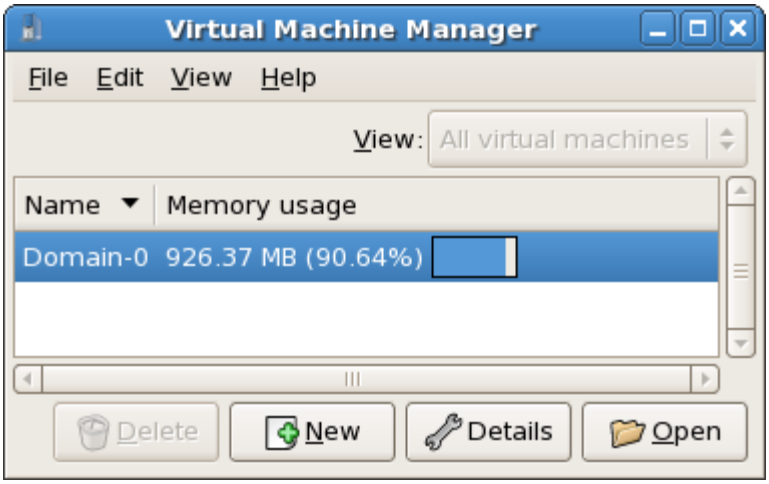

图 **18.38.** 显示内存使用情况

第 **19** 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。

19.1. 日志文件概述和位置

当在你的网络里部署带有虚拟化的红帽企业 Linux 5.0 时，主机的虚拟化软件使用许多特殊的目录来容纳配置文件、日志文件和其他工具。所有红帽虚拟化系统的日志文件都是标准的 ASCII 文件，可以很容易地用基于 ASCII 的编辑器存取。

- 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。

19.2. 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。

19.3. 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/

19.4. 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：

- XenOprofile
- 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
- brtctl

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.feffffff       no        vif13.0

xenbr1             8000.ffffefff       yes       pddummy0

xenbr2             8000.ffffffef       no        vif0.0

# btcrl 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

# btcrl showstp xenbr0

xenbr0
```

```
bridge-id               8000.feffffffff

designated-root         8000.feffffffff

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
```

19.5. 利用日志进行故障解除

如果在安装红帽虚拟化系统时遇到问题，你可以参考主机系统的两个日志文件来协助接触故障。

xend.log 文件包含了于你运行 xm log 命令相同的基本信息。它位于 /var/log/ 目录里。下面是一个当你创建 运行内核的域时的示例日志项:

```
[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。

19.6. 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 Linix (2.618-8.2080_RHEL5xen0)

                root (hd0,2)

                kernel /xen.gz-2.618-8.el5 com1=38400,8n1

                module /vmlinuz-2.618-8.el5xen ro root=LABEL=/rhgb quiet console=xvc

console=tty xencons=xvc

                module /initrd-2.618-8.el5xen.img
```

sync_console 可以帮助你诊断导致异步的监控程序控制台输出挂起的问题，"pnpacpi=off" 可以绕过中断串行控制台输入的问题。参数 "console=ttyS0" 和 "console=tty" 表示内核错误将在普通的 VGA 控制台以及串行控制台上被记载。然后你可以通过标准的非调制解调器（null-

modem）的电缆来安装并设置 ttywatch ，捕获远程主机上的数据。例如，在远程主机上，你可以输入：

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

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

19.7. 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`。

19.8. 对完全虚拟化客户机控制台的访问

完全虚拟化客户机操作系统自动使用一个配置好的文本控制台，但是区别是内核客户机没有被配置。要使客户机虚拟串行控制台和完全虚拟化客户机一起使用，你必须修改客户机的 grub.conf 文件，并包含 'console =ttyS0 console=tty0' 参数。这确保了内核信息被送至虚拟串行控制台（和普通的图形化控制台）。如果你计划在完全虚拟客户机内使用虚拟串行控制台，你必须编辑 /etc/xen/ 目录里的配置文件。在主机域上，你可以键入下列命令来访问文本控制台：

```
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。

19.9. 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

*3600a0b8000132750000015427b625e*
```

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="3600a0b8000132750000015427b625e

", 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       3600a0b8000132750000015427b625e

              alias      oramp1

}

multipath  {

              wwid       3600a0b8000132750000015427b6

              alias      oramp2

}

multipath  {

              wwid       3600a0b8000132750000015427b625e

              alias      oramp3

}
```

```
multipath  {

              wwid        3600a0b8000132750000015427b625e

              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。

19.10. 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。

19.11. 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 VolGroup00now

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/

#vchange -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。

19.12. 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）。

19.13. 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。、

19.14. 客户机创建错误

当你试图创建客户机时，你遇到 "Invalid argument" 错误信息。这通常是代表你试图引导的内核映像和监控程序（hypervisor）不兼容。例如，当你试图在支持 PAE 的 FC6 hypervisor 上运行 non-PAE FC5 内核时，就会出现这个问题。

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。

19.15. 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-2.8.6.18-1.2910.el5xen.img
```

```
title RHEL5 i386 xen (2.618.-1290.el5xen

root (hd0, 8)

kernel /boot/xen.gz-2.618-1290.el5 com2=115200 console=com21

module /boot/vmlinuz2.618-1290.el5xen to root=LABEL=RHEL5_i386 console=xvc xencons=xvc

module /boot/ititrd-2.618-1290.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 。

19.16. 网桥错误

红帽虚拟化系统可以配置多个虚拟化网桥来与以太网卡一起使用。要成功地为以太网卡配置多个网桥，你必须用 system-config-network TUI/GUI 程序来配置第二个网络接口，或者在 /etc/sysconfig/network-scripts 里创建一个新的配置文件。你应该使用一个进程来设置多个 Xen 网桥。下面是用于第二个网卡 'eth1' 的配置示例文件：

```
#/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=10.11.1.1

GATEWAY=10.11.254

ARP=yes
```

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

编辑 /etc/xen/xend-config.sxp 并在新的网桥脚本里加入一行（这个例子使用了 "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。

19.17. 笔记本配置

配置装有 RHEL 5.0 的笔记本在网络环境里使用是一项具有挑战性的任务。大部分系统都经常切换 WiFi 和有线连接，而红帽虚拟化系统假定一直访文同一接口。这导致系统对红帽红帽虚拟化系统使用的网络接口执行 ifup/ifdown 操作。既然红帽虚拟化系统使用缺省的网络接口，所以 WiFi 卡不是理想的网络连接方法。

解决办法就是创建一个专门给红帽虚拟化系统使用的 'dummy' 网络接口。

这个技术允许你为客户机和虚拟机使用一个隐藏的 IP 地址空间。要实现这种方法，你必须使用静态 IP 地址，这是因为 DHCP 不会侦听 dummy 网络上的 IP 地址。你也必须配置伪 NAT/IP 来启用对客户机和虚拟机的网络访问。当你创建 'dummy' 网络接口时，你应该附加一个静态 IP。

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。

你将需要在 xend-config.sxp 文件里进行其他的修改。你必须定位（network-script 'network-bridge bridge=xenbr0 ）部分并在行末添加：

```
netdev=dummy0
```

你也必在客户机的 domU 网络配置里进行一些修改来启用指向 dummy0 的网关。如下例所示，你必须编辑位于 /etc/sysconfig/ 目录里的 DomU 'network' 文件：

```
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/

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
```

19.18. 在系统引导时自动启动域

在系统引导时自动启动域

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]#
```

19.19. 修改 Domain0

要用红帽虚拟化系统来管理 domain0，你将需要修改 /etc 目录里的 grub.conf 文件。因为要管理大量的域，在编辑 grub.conf 时，许多系统管理员宁愿使用“复制/粘贴”的方法。如果你也采用这个方法，请确保你包括了虚拟化条目里的全部五行（否则会导致系统错误）。如果你需要与 Xen hypervisor 相关的值，你必须把它加到 'xen' 行里。下例代表了 grub.conf 文件里的一个正确的虚拟化条目：

```
# 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-2.6.17-1.2519.4.21.el5xen ro root=/dev/VolGroup00/LogVol00

        module /initrd-2.6.17-1.2519.4.21.el5xen.img
```

例如，如果你需要把 dom0 hypervisor 引导时的内存修改为 256MB，你必须编辑 'xen' 行并把附上正确的条目：'dom0_mem=256M' 。下面的例子描述了不同的 grub.conf 里的 xen 条目：

```
# 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 (2.6.17-1259.4.21 el5xen)

        root (hd0,0)

        kernel /xen.gz-2.6.17-1259.4.21el5 com1=115200, 8n1 dom0_mem=256MB

        module /vmlinuz-2.6.17-1259.4.21el5xen ro

        root=/dev/VolGroup00/LogVol00

        module /initrd-2.6.17-1259.4.21el5xen.img
```

19.20. 客户机配置文件

当你用红帽企业 Linux 5.0 里的 virt-manager（或 virt-install）安装新的客户机时，客户机配置文件（位于 /etc/xen 目录）将被修改并自动被设置。下面是半虚拟化客户机的一个配置文件示例：

```
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"
```

注意，serial="pty" 是配置文件的缺省设置。下面是完全虚拟化客户机的一个配置文件示例：

```
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 = "b1037f9-91d7-ao5f-12ff-3721c0c99af5'

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'
```

19.21. 克隆客户机配置文件

你可以复制（或克隆）现有的配置文件来创建新的客户机。你必须修改客户机配置文件的 name 参数。然后这个新的、唯一的名称将出现在监控程序里并对于管理工具可见。你也必须生成所有新的 UUID（使用 uuidgen(1) 命令）。对于 vif 条目，你必须为每个客户机定义一个唯一的 MAC 地址（如果你是复制现有的配置文件，你可以用一个脚本来实现）。对于 xen 的网桥信息，如果你把现有的配置文件移动到一个新的主机，你必须更新 xenbr 条目来符合你的本地网络配置。对于 Device 条目，你必须修改 'disk=' 部分的条目来指向正确的客户机映像。

你也必须修改客户机的这些系统设置。你必须修改 /etc/sysconfig/network 文件里的 HOSTNAME 条目来匹配新客户机的主机名。

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，你必修改 IPADDR 条目。

19.22. 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
```

19.23. 配置虚拟机的 Live 移植

红帽虚拟化系统可以在其他运行带有虚拟化功能的红帽企业 Linux 5.0 的服务器间移植虚拟机。而且，可以采用离线（offline）移植方式（使用 xm migrate 命令）。Live 移植可以用相同的命令来完成然而，你还得对 xend-config 配置文件进行一些其他的修改。下面的例子列出了进行成功移植你必须修改的条目：

(xend-relocation-server yes)

这个参数的缺省值是 'no'，它使 relocation/migration 服务器保持无效状态（除了信任的网络），而域虚拟内存以不加密的原始方式进行交换。

(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)

这个参数控制与 relocation 端口通信的主机。如果这个值是空值，表示允许所有的转入连接。你必须把它修改为用空格隔开的常规表达式（如xend-relocation-hosts-allow-'^localhost\\.localdomain$'）。匹配这些常规表达式的全限定域名或 IP 地址都被接受。

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。

19.24. 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
```

通过 xm list Domain0 命令，你可以检查 domain0 所使用的内存数量。如果 domain0 已经没有可用内存，你可以用 "xm mem-set Domain-0 NewMemSize" 来设置新的内存容量。

你接收到下面的错误：

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

这个信息表示你试图在监控程序里运行不被支持的客户机内核映像。当你试图在 RHEL 5.0 监控程序里引导 non-PAE 半虚拟化客户机内核时，将产生这个问题。红帽虚拟化系统只支持带有 PAE 的客户机内核和 64 位的体系结构。

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]
```

```
XendDomainInfo.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。

当你把完全虚拟化的 HVM 客户机移动到 RHEL 5.0 系统里时，就会出现这个问题。你的客户机不能够引导且在控制台可以看到一个错误信息。检查配置文件里的 PAE 条目，确保 pae=1。你应该使用 32 位的版本。

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!
```

当客户的网桥没有正确配置时会产生这个问题，它会迫使 Xen hotplug 脚本超时。如果你在主机之间移动配置文件，你必须确保你更新了这些配置文件，如进行网络拓扑结构和配置的修改。当你试图启动含有不正确或不存在的 Xen 网桥配置的客户机时，你将接收到如下的错误：

```
[root@trumble virt]# xm create r5b2-mySQL01



Using config file " r5b2-mySQL01"

Going to boot Red Hat Enterprise Linux Server (2.6.18.-1.27 .el5xen)

kernel: /vmlinuz-2.6.18-1.27.el5xen

initrd: /initrd-2.6.18-1.27.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。

19.25. 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 20 章 其他的资源

要学习更多关于红帽虚拟化的知识，请参考下面的资源。

20.1. 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，红帽虚拟化源于这个项目。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。

20.2. 安装的文档

- /usr/share/doc/xen-*<version-number>*/ —. 这个目录包含了关于 Xen 半虚拟化 hypervisor 以及相关管理工具的大量信息，包括不同的示例配置、和硬件相关的信息以及当前的 Xen 用户文档。
- man virsh and /usr/share/doc/libvirt-*<version-number>* — 包含了 virsh 虚拟机管理工具的子命令和选项，以及关于 libvirt 虚拟化库 API 的综合信息。
- /usr/share/doc/gnome-applet-vm-*<version-number>* — 监控和管理本地运行的虚拟机的 GNOME 图形化面板 applet 的文档。
- /usr/share/doc/libvirt-python-*<version-number>* — 提供 libvirt 库的 Python 绑定的细节。libvirt-python 软件包允许 python 开发者用 libvirt 虚拟化管理库编写程序。
- /usr/share/doc/python-virtinst-*<version-number>* — 提供在虚拟机里启动 Fedora 和红帽企业 Linux 安装的 virt-install 命令的文档。
- /usr/share/doc/virt-manager-*<version-number>* — 提供虚拟机管理者（管理虚拟机的图形化工具）的文档。

附录 A. Revision History

修订历史	
修订 5.0.0-8	Thu Apr 05 2007
Resolves: #235311 Clarifying SELinux installation procedure	
修订 5.0.0-7	Wed Feb 07 2007

Resolves: #224220 #225169 Additional Developer Feedback	
修订 5.0.0-6	Thu Jan 31 2007
Resolves: #224220 #225169 Modify troubleshoot command	
修订 5.0.0-3	Thu Jan 11 2007
Resolves: #221137 Fix to broken rpm	

附录 B. 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：

```
2.  grep pae /proc/cpuinfo

3.
```

4. 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。

```
5.  flags :

6.      fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat clflush dts
    acpi
7       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 所为你的虚拟机名称。
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. 启动红帽虚拟化管理者应用程序，在 `File` 菜单里，点击 `New`。
4. 点击 `Forward`。
5. 键入 rhel5b2-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/rhel5b2 作为安装介质的 URL，然后点击 `Forward`。
8. 选择 Simple File，键入 /xen/rhel5b2-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. 要知道你的 CPU 是否支持 Intel-VT 或 AMD-V，键入下面的命令： egrep -e 'vmx| svm' /proc/cpuinfo
2. 下面的输出表示 CPU 支持 Intel-VT：

```
3.  .flags :

4.      fpu tsc msr pae mce cx8 apic mtrr mca cmov pat clflush dts acpi mmx fxsr
    sse

5.      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。

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

```
at /sys/hypervisor/properties/capabilities
```

7. 下面的输出表示 BIOS 已经启用了 Intel-VT 支持。如果这个命令没有任何输出，你可以在 BIOS 设置工具里寻找和虚拟化相关的设置，如 IBM T60p 里的 'CPU' 部分里的 'Intel(R) Virtualization Technology'。启用这个选项并保存，然后重启机器来使它生效。

```
8.  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 rhel5b2-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/rhel5b2-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. VNC viewer 将在安装窗口里出现。如果有类似于“main: Unable to connect to host: Connection refused (111)”的错误信息，可以键入下面的命令来继续： vncviewer localhost:5900。VNC 端口 5900 对应的是运行在 VNC 上的第一个 Xen 客户。如果没有成功，你可能需要使用 5901、5902 等。

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. 启动红帽的虚拟机监控器应用程序，并在 `File` 菜单里，点击 `New`。
4. 点击 `Forward`。
5. 键入 rhel5b2-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/rhel5b2-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. 启动红帽的虚拟机管理者应用程序，然后在 File 菜单里点击 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. 任何时候，只要出现 'Files Needed' 对话框，你可以把路径 GLOBALROOT\DEVICE\CDROM0\I386 修改为 C:\I386。你也可能不会看到这个问题，这依赖于你的安装。在安装过程中你可能被提示有文件丢失。把路径修改为 C:\I386 应该可以解决这个问题。

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=1sdl=1#vcpus=2 。
19. 重复步骤 14 并象平常一样进行安装。

附录 C. 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
```

4. 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$')
```

10. 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. 创建 /home/mp3 目录并从红帽的 Truth Happens 页里复制 TruthHappens.ogg：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

```
4.  command:gnump3d
```

5. 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。
6. 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：

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

4.
```

5. 用下面的命令在两台 Xen 主机上打开多个窗口终端：

```
6.  watch -n1 xm list
```

7. 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

如果时间允许的话，在 Xen 客户机里，配置 VNC 服务器在 gdm 启动时初始化。运行 VNC viewer 并连接至 Xen 客户机。当 live 移植发生时测试 Xen 客户机。试图暂停/恢复和保存/恢复 Xen 客户机并观察 VNC viewer 里发生了什么。如果你通过 localhost:590x 连接至 VNC viewer，并开始 live 移植，当移植失败时你将不能够再次连接到 VNC viewer。这是一个已知的程序错误。