

开源时代

OPEN SOURCE TIMES

2009年5-6月合刊 总第九期

技术新知:

基于Nessus的主机网络安全检测
具有负载均衡功能的MySQL服务器
集群部署及实现

开源业界:

普通桌面操作系统领域 Linux难成大业
开发者眼中的Moblin和Android

社区扫描:

MySQL被分支已无法挽救?
是Apache还是GPL更有利于开源商业?

行业观察:

谁在打造Android的技术基石?
Red Hat被收购不可避免 IBM是最可能买家

专家专栏:

参与社区开源项目 贵在坚持

——Novell中国区研发经理刘俊贤专访

国内开源社区发展总是风雨不断,除了部分开源用户社区发展还算良好外,优秀的开源项目社区一直很缺乏。今天我们有幸请到 Novell中国区研发经理刘俊贤先生,从个人爱好和工作两个角度,来和我们谈谈如何在国内投身开源软件事业。





2009年5-6月合刊 总第九期

编辑出品: 

网络发行: ChinaUnix

主编: 江晖
技术主编: 樊强
执行编辑: 周荣茂
内容编辑: 周荣茂 覃里
唐川 周平
技术编委: 高延斌 马路遥
白金
美术编辑: 林在子
交流论坛: bbs.chinaunix.net
杂志下载: www.chinaunix.net
www.itpub.net
www.ixpub.net

联系我们:
rmzhou@staff.chinaunix.net
投稿邮箱:
rmzhou@staff.chinaunix.net

媒体支持: 

广告联系: 温玉琴
电话: 010-82658790
手机: 13801339139
E-mail: wyq@it168.com

内容目录

卷首语

004

开源业界

Linux操作系统实际市场占有率	005
Intel诺基亚合作Linux手机系统 值得期待	005
普通桌面操作系统领域 Linux难成大业	006
外刊盘点不用Linux七大理由: 社区成员不友好	007
Ubuntu创始人:Linux的未来并不在于对Windows的兼容性	008
惠普推廉价ProBook商务笔记本 首次预装Linux	008
Acer宣布开始Android开发	009
Intel与Novell合作推广Moblin操作系统	009
开发者眼中的Moblin和Android	010
红帽EMEA合作伙伴峰会结束 参会人数上升40%	011
Linux基金会:这一次微软是我们同床异梦的战友	012
非洲上演开源之战	012
联通宣布加入开放手机联盟 将开发手机操作平台	013
十个优秀的特征 让你喜欢上Android 1.5	014
Novell张先民: Linux发展中心渐转向中国	016

社区扫描

Cisco将会召集到下一代Linux开发人员?	018
MySQL创始人宣布创建开源数据库联盟	018
OpenOffice前途惨淡, 我们还有其它选择吗?	018
是Apache还是GPL更有利于开源商业?	022
开源社区争论话题之三: 使命何在?	023
Ubuntu 9.10: 在性能飞跃中昂首起步	025
MySQL被分支已无法挽救?	030

行业观察

谁在打造Android的技术基石?	031
开源存储的现状	032
Red Hat被收购不可避免 IBM是最可能买家	034
MINIX 3承诺比Windows或Linux更安全	035
开源技术: 经济衰退期企业IT变革之道	036

专家专栏

参与社区开源项目 贵在坚持	038
---------------	-----

技术新知

基于Nessus的主机网络安全检测	042
具有负载均衡功能的MySQL服务器集群部署及实现	051
红帽RHEL5U3平台实现pppoe接入记账认证服务器	065
根文件系统所在LVM无损收缩全攻略	070
实现Linux软件RAID的关键技术讨论	078
VMware 环境下OpenBSD 4.5以及Gnome 2.24的安装	084
Part I OpenBSD 4.5的安装	084

网友热评

109

版权声明

杂志内容来自ChinaUnix社区及互联网, 电子杂志的宗旨是为了更好地传递开源最新自寻和技术经验。如有版权问题敬请联系, 我们将会第一时间做出处理。

致谢

本杂志得到ChinaUnix网站Linux时代社区版主的大力支持, 技术文章大部分来自版主推荐, 更多技术文章可以访问Linux时代精华区。本刊分析评论部分文章来自IT168技术频道。

卷首语

五月即将过去，新的一期杂志又和大家见面了。时间过的很快，《开源时代》杂志第九期到来，CU 论坛八周年庆也即将到来！为了配合论坛八周年庆，本期杂志将推出有奖下载活动，给一直支持我们的读者一些小奖励！另外我们将为即将到来的《开源时代》杂志周年庆举办一个有奖投票活动，请大家给即将周岁的《开源时代》提意见，没有大家的关爱，《开源时代》也不会走到今天！

过去的一个月里开源业界依旧纷纷扰扰，特别是在 Linux 市场占有率上。NetApplications 最近的调查显示，Linux 操作系统在桌面操作系统市场已经突破了 1%。但是立马有人跳出来反对，认为 Linux 只是适合服务器操作系统市场，桌面操作系统市场占有率没有意思，但是无论如何，Linux 在桌面操作系统市场步履蹒跚也是事实，无论是厂商、用户还是整个生态圈的建设方面都是存在问题的，如何不能取得突破，光靠硬件厂商预装和一小部分的爱好者的需求，Linux 桌面操作系统市场堪忧。

与桌面市场迟迟不能打开局面相比，Linux 操作系统在嵌入式系统市场却是突飞猛进，特别是在开放式手机操作系统上。也许 Intel 是受到 Android 对上网本市场蠢蠢欲动的的影响，最近联合 Nokia 推出 oFono 平台，开始和 Android 争夺开放手机操作系统市场。再加上原有的 OpenMoko、和 LiMo 等开放式手机平台，目前这个市场已经高手如云。

制衡和反制衡，向来是竞争对手之间永恒的两个词。就像上面提到的 Google 的 Android 平台和 Intel、Nokia 推出的 oFono 平台。Oracle 和 IBM 的红蓝大战依然在继续。虽然 Oracle 收购 Sun 还没有尘埃落定，但是 IBM 在 Java 方面的被动之势已经开始显现，暂时处于被动的 IBM 也在寻找出路，也许红帽确实是他们一个好的选择，红帽旗下的 JBoss 不失为一款非常具有竞争力的开源应用服务器解决方案，对于 IBM 现有的 webspheres 是一个绝佳的补充。

就在三月份 Novell 发布最新 Novell SUSE Enterprise 11 的时候，中国区总裁还否认 Novell 将参与到上网本 Linux 操作系统市场，但是在四月份的时候，Novell 隆重宣布和 Intel 公司 Moblin 项目合作，而这一切仅在 Moblin 转交 Linux 基金会的一个月后。

三月初的时候，Linux 基金会买下了 Linux.com 域名，有意将其建设为 Linux 用户和开发者的活跃社区，有分析人士认为，Linux 基金会此举是为了吸引更多的商业用户使用 Linux 操作系统。就在五月初的时候，Linux.com 网站全新上线，并且接管 Moblin 项目，不仅代表了 Linux 基金会的强势上升，更是说明 Linux 在业界不断受到重视。

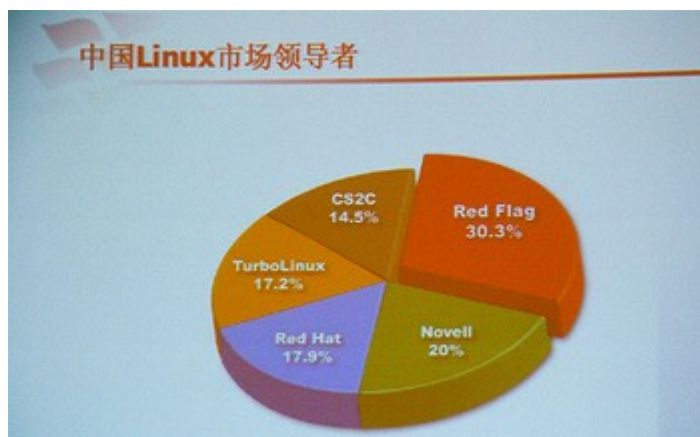
Cu 技术沙龙正在朝着良性的方向发展，就在我写下这段文字的同时，我们的两位同事正在飞往广州的途中，开源网络性能调优沙龙将如期在广州举行。而在过去的这个五月里，来自古城西安的 CU 网友也自发举办了一场 Linux 技术沙龙，西安本地的网友非常积极地主办了这次西安 Cuer 的聚会，非常感谢他们的辛苦组织！在六月份，ChinaUnix 技术沙龙将迎来繁忙的一个月，我们将在北京、西安和成都三地举办三场技术沙龙，主题涉及到虚拟化、网络架构优化和高性能计算等。如果不出意外的话，很有可能和成都的 CU 网友在沙龙会场见面！

周荣茂

开源业界

Linux 操作系统实际市场占有率

Net Applications 近日公布了 4 月份操作系统市场份额统计报告，其中 Linux 份额突破 1%，达到 1.02%。但开源支持者对此结果表示愤慨，并声称 Linux 的市场份额实际上在 10% 左右。



鉴于两个数字相差悬殊，Linux 支持者 Bruce Byfield 引用多方数据对 GNU/Linux 用户数量进行了估算，最后得出的结论是 Linux 市场份额在 2% – 5% 之间。

据 Bruce Byfield 分析，这个数字难以精确计算的原因在于 GNU/Linux 来自于几十个供应商和社区项目，他们大多不销售操作系统光盘、也不需要在线激活。在这种情况下估算用户数量并不容易。NetApplications 的统计结果实际上是一种抽样调查结果，这种调查方式对占市场份额较小的样本存在较大的误差，GNU/Linux 的用户可能较多选择了自由软件网站，而没有访问 NetApplications 设定的检测网站。

此外 NetApplications 是一家美国公司，并没有海外分支机构，所以统计数据仅仅是从世界各地访问美国网站的结果来进行抽样的，鉴于此，NetApplications 的数字并不能表现实际的 Linux 用户量。

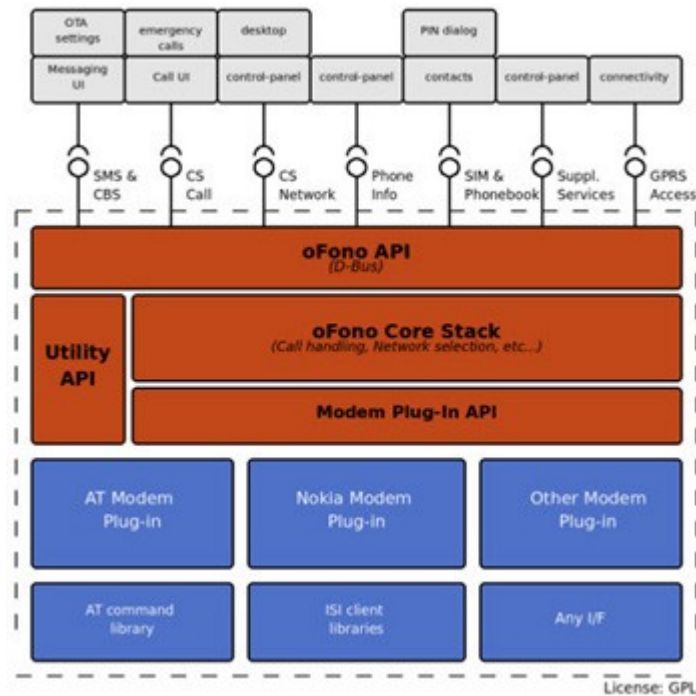
最后 Bruce Byfield 使用 Mozilla 公司雇员提供的数字来统计 Linux 用户量，其中原因在于 Firefox 是 75% – 90% Linux 系统中的默认浏览器。Mozilla 的数据显示，Firefox 安装量在 2.5 亿左右，其中 2 – 5% 为 Linux 用户。鉴于并不是所有 GNU/Linux 用户都使用 Firefox，所以这个数字可能比此数值要高。所以，Linux 的实际用户量肯定高于 1%。

Intel 诺基亚合作 Linux 手机系统 值得期待

英特尔是全球半导体芯片龙头，诺基亚则是全球手机龙头，乍一看两家公司之间似乎没有太多交集可言，但近日却有消息曝出这两家大腕级厂商正在精诚合作，看来我们又有精彩的故事可看了。近日有消息透露，英特尔和诺基亚正在着手合作推出基于 Linux 内核的手机专用操作系统。双方合作的手机操作系统项目称作“oFono 计划”，“oFono 计划”独立于双方平时的 Linux 项目，英特尔和诺基亚有意通过“oFono 计划”打造出一个类似 Android 的开放源架构手机平台。

英特尔和诺基亚合作的 Linux 手机操作系统本身功能将会非常有限，但是却拥有强大的可扩展性，

因此下游合作公司将可以自主加入自己的网络和个性功能。目前 oFono 计划暂未透露该 Linux 操作系统是否会对智能手机有用，但是有消息透露，一家很可能与诺基亚有关的公司 Phonedog 正在搜索熟悉 OpenGL 标准的图形界面人才，而 OpenGL 标准可以适用于 2D、3D 以及诺基亚的 Qt 图形界面应用程序架构。



不过从另外一个角度来看英特尔和诺基亚对合作所谓的“oFono 计划”都缺乏足够的动机。首先诺基亚已经是 Symbian 的第一大股东，并且自去年以来已经将 Symbian 平台打造成了一个既独立又逐步开源的系统；况且诺基亚本身也已经推出过基于 Linux 的 Maemo 系统，只不过该系统只适用于 N810 等较大设备而已。英特尔方面虽然也有 MoblinOS，但是同时英特尔也在为杀入移动互联网设备及其他较大的计算设备而努力。诚然，如果双方合作 Linux 操作系统的话可能会与现有的 Maemo 或者 Moblin 有所冲突，但如果从整合资源的角度来看的话英特尔和诺基亚就基于 Linux 的操作系统展开合作也算合情合理，至于真实的情况如何还是等待时间来证明吧。

普通桌面操作系统领域 Linux 难成大业

NetApplications 最新表示，Linux 的应用已经突破了 1% 的市场份额，虽然这是 Linux 的一个重要里程碑，但是，Linux 可能永远不会成为一个桌面操作系统的重要参与者。

Linux 作为服务器操作系统一直是非常的成功的，但是，在普通的 PC 市场上，一直没办法追赶在服务器市场的份额，普通的客户机突破 1% 是因为 Linux 已经有非常多的应用功能，并且预装到戴尔等计算机厂商的电脑中。

StevenVaughn-Nichols 认为，Linux 最终将占市场份额的 10%~20%，但是，与分析师认为，这个预测有点不切实际，他们认为，Linux 的市场份额不会超过 5%，甚至 2% 都将是非常困难。

分析师认为，Linux 虽然在服务器市场有着不错的发挥，但是，如果在普通的 PC 市场，我们几乎很难察觉到他的存在，虽然 Linux 使用越来越容易，但是，他有着太多的变体，当我们想在 Linux

上安装应用软件时，就会发现，太困难了。

Linux 会突破 1% 的市场份额吗？目前上网本中，大概有 30% 的操作系统使用的是 Linux，因此，如果 Linux 要在台式电脑中有一定得发展，太困难了。

外刊盘点不用 Linux 七大理由：社区成员不友好

导读：知名 IT 杂志《PC World》网络版 5 月 1 日载文称，许多计算机用户都曾尝试过 Linux，但大多数都因各种各样的原因例如找不到合适软件等而重新投入到 Windows 的怀抱。以下为用户放弃 Linux 的七大理由：

1、Linux 不能运行某些常用软件

计算机用户经常抱怨 Linux 不能运行 Adobe 的某些软件例如 Photoshop 或 Dreamweaver，而且找不到可替代的开放源代码产品。最近，桌面 Linux 在浏览器和办公软件领域取得了长足进展，但在一些专门软件领域仍然有所欠缺，例如高质量的 Web 设计软件。

解决这一问题很简单：如果必须使用一些不能在 Linux 上运行的专门软件，那就继续使用 Windows 好了。

2、Linux 不支持某些硬件

某些硬件可能无法在 Linux 下正常运行，例如显卡和 WiFi。但是，网上有许多介绍解决此类问题的资料，用户可以通过查阅资料解决出现的问题，这也是一种乐趣。

3、只能使用命令行交互方式

许多 Linux 功能确实只能通过命令行完成，但使用键盘输入命令对于普通用户而言并非什么难事。这或许是使用 Windows 操作系统惯出来的一个毛病。

4、在许多方面与 Windows 不同

从来没有人认为 Linux 是 Windows 的克隆品。Linux 与 Windows 没有高下之别，只是两者不同而已。用户需要适应 Linux。

5、在论坛上发帖后，Linux 社区成员不友好

所有社区的成员都不是完人，这不是 Linux 的错。

6、我不喜欢 Linux

Linux 并非适合所有计算机用户，在可用性方面不如 Windows 或 OS X。但许多用户因为对 Linux 不熟悉才有了这种看法，于是重新使用 Windows。

7、Linux 带来了许多问题

许多持这种看法的用户都会说，在安装 Linux 时，安装软件中途出了问题。我试图重新启动系统，但系统进入了命令行状态。即使显示出了桌面，软件也不能正常运行。

这只是个例，并非大多数人都会有这种遭遇，只能怪自己运气差。

Ubuntu 创始人:Linux 的未来并不在于对 Windows 的兼容性

Ubuntu 的创始人 Mark Shuttleworth 最近对桌面 Linux 以及其未来发表了一些比较重要的言论，他说 Linux 的未来不在对 Windows 的兼容性上面。在其发言中，他特意提到了 Wine 这个程序——这是一个可以在 Linux 上提供对 Windows 程序的支持的程序，如果要想从 Windows 迁移到 Linux，而你有些程序无法割舍，使用它再好不过，而且很多时候你会发现你想要的程序在 Windows 下才有。

Mark Shuttleworth 称，类似 Wine 的兼容 Windows 的程序或者工程其重要程序不言而喻，但 Linux 的未来不在于对 Windows 的兼容性。Linux 与 Windows 有着本质的区别。linux 作为自由软件，这个平台是否成功要依靠自身的发展，如果使用 linux 只是一种运行 Windows 程序的媒介，那么 linux 永远不会有成功的那一天。

虽然他仅仅针对 Ubuntu 发表以上言论，但是很显然这番言论对任何 linux 版本都是合适的。正如他所说，在 linux 模拟运行 Windows 程序并不能很好的抢占这个市场。linux 开发者应该把重点放在核心应用上，让 linux 也可实现所有 Windows 才可实现的功能。

惠普推廉价 ProBook 商务笔记本 首次预装 Linux

近日，据路透报道，惠普公司针对中小企业用户，推出廉价新型商务 ProBooks 系列笔记本电脑。该系列新产品定于周二全球出货，继去年 EliteBook 系列高端轻薄本之后，将成为惠普新的主流商务本。

ProBook 本推出了一些新功能，包括可选择预装 Linux 操作系统，为那些不想使用 Windows 操作系统的用户提供方便。这是惠普首次在标准商务笔记本上推出 Linux 预装服务。ProBook 笔记本屏幕尺寸分 14 寸、15.6 寸和 17.3 寸三种，起价为 529 美元。该系列中另一个惠普商务本新特色是，用户可以选择对笔记本外壳进行色彩处理，添加“merlot”葡萄酒红色。



一些型号 ProBook 还添加了高通 Gobi 技术，装一个模块即可使用不同的移动宽带网络技术和移动运营商。该系列电脑也兼具惠普高端笔记本的耐用特点，引入了惠普的硬盘保护技术 3D DriveGuard，以及防泼水键盘。

Acer 宣布开始 Android 开发

Acer 面对智能手机市场一直毫不手软，在 2008 年年底收购倚天后就开始不断推出新品，在最近举行的电话会议上，Acer 宣布他们对 Android 手机非常感兴趣，在不久的将来就可能会有产品推出。

在台北举行的公司第一季度投资者会议上，Acer 总裁兼 CEO Gianfranco Lanci 表示，Android 已经在不同的几个 Acer 手机硬件解决方案上开始测试。

Acer 的开发人员们正在努力工作，也不排除推出基于 Andorid 系统的 Netbook 的可能。

在 3 月底时，Acer 曾经被问到是否将进入 Android 开发领域，但那时他们一直拒绝对此发表评论，现在这一悬念终于揭开，预计开发将持续一段时间

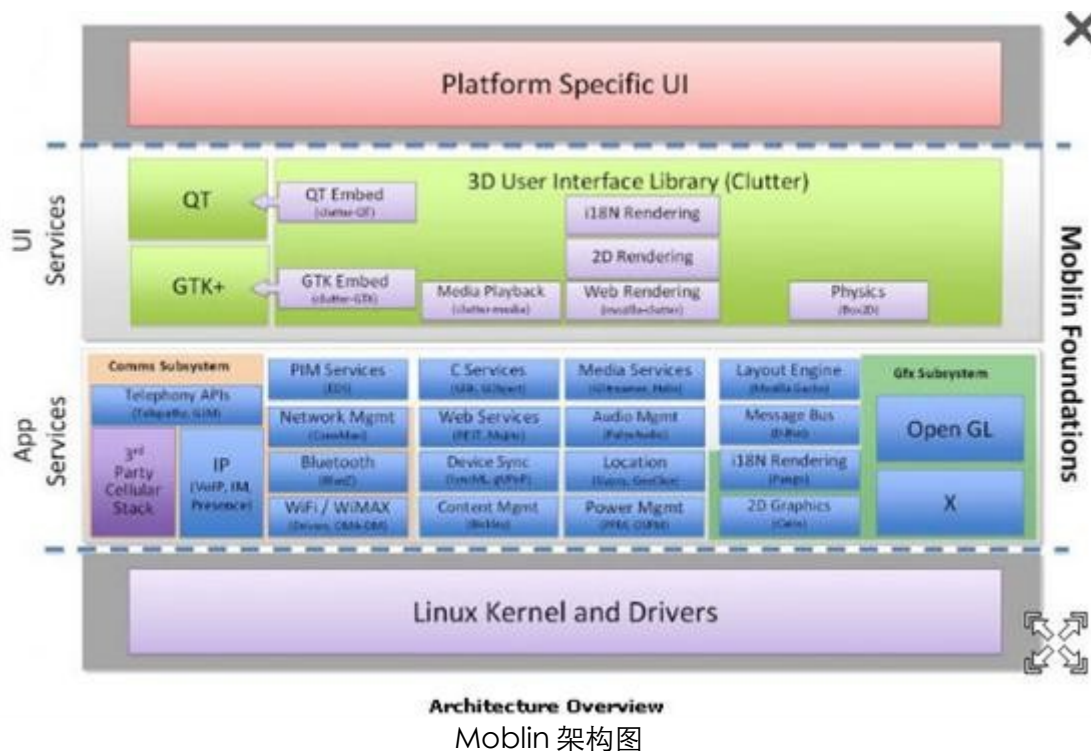
Intel 与 Novell 合作推广 Moblin 操作系统

Intel 和 Novell 昨日宣布将开展合作，共同推广开源 Linux 操作系统 Moblin，这是 Intel 自 2007 年开始就推出的一款操作系统，并在上月初将其移交给了 Linux 基金会。

Novell 总裁 Ron Hovsepian 表示，Intel 和 Novell 计划通力合作，鼓励硬件厂商采用 Moblin 操作系统，Novell 也准备面向上网本开发一款基于 Moblin 的产品。Hovsepian 称，除了开发基于 Moblin 操作系统的相关产品，Novell 还将在台湾成立实验室，进一步推动 Moblin 的普及，并与台湾 Moblin 技术发展中心（Moblin Enabling Center）共同研究基于 Moblin 的硬件设计。

Moblin 是 Linux 操作系统的一个发行版，由 Intel 在 2007 年发布，Novell 在去年十月开始加入 Moblin 的开发，主要贡献在于窗口、Email 和 Media 管理代码。

当谈到另一款 Linux 系统 Google Android 时，Intel 首席 Linux 和开源软件专家 Dirk Hohndel 表示，Android 和 Moblin 以不同的方式使用 Linux，Moblin 主要面向上网本，两者有很大区别。



开发者眼中的 Moblin 和 Android

就目前的市场形式分析，Moblin 更倾向于带数据互联功能的移动设备，而 Android 则更倾向于带电话功能的移动互联设备。就系统成熟度来说，Moblin 稍胜一筹，从目前的终端基数和应用的多样性来看，Android 则更具优势，拥有成熟的 Online App Market 系统，并且每天都有数款软件被发布。

下面，我就为大家分析一下 Moblin 和 Android 的不同。

“Moblin 项目由 Intel 公司于 2007 年 7 月正式启动，最初主要集中于开发 MID 软件系统，基于 Linux* 的操作系统作为突破口出现了新的产品设备种类，能够访问 Internet 上丰富的内容，同时没有与 PC 相关的复杂性。这些设备从移动互联网设备（Mobile Internet Devices, MID）、车载信息系统、便携式媒体播放器（PMP）到个人导航设备（PND）、数码相机顶盒（STB）到笔记本电脑等以各种形式推动新技术、新设备的发展。”

从这段表述中我们可以很容易的获知相对于 Android 主要支持的 ARM 架构，Intel 的 Moblin 初期所面向的是以移动 PC 为技术基础的那些制造商，简而言之，是面向 Intel Atom 芯片家族的移动型操作系统。Moblin 的概念来自移动设备制造商想要一个优化过的移动用户操作界面，主要以触摸或指点杆为基础的出入方式，拥有高速的联网性能，占用资源小的优化过的移动操作系统。

Moblin v2 作为 Moblin 的最新版本，更多的是对核心堆栈进行了优化，并且针对移动平台最重要的电源管理部分进行大幅度优化，使得续航时间得到大大加强。

在互联网方面，Moblin v2 开始支持更多的数据互联技术，使得较上网本更轻量一级的 MID 3G 上网设备成为可能，并且通过 3G 等宽带技术获得丰富的富媒体信息。

Moblin v2 为设计主屏幕和图形界面有了以下四种选择：Flash 界面：以 Adobe AIR 平台作为基础，使操作系统脱离浏览器运行 Flash，开发者可直接用 Flash 作为程序的图形界面接口设计丰富多彩的动态图形界面。Html 界面：直接以 Html 文件作为操作系统主屏幕或者应用程序图形界面，开发者可设计具备互联网访问能力的各种图形界面。Clutter 界面：Clutter 是以 OpenGL 作为基础的 3D 动画图形库，提供面向对象的简单易用接口。该库将 3D 动画应用到基于 2D 布局的程序界面上获得丰富的 3D 特效和更为出色的性能。GTK 界面：GTK 是 Linux 系统中主流的图形界面接口，接口丰富，性能良好且开发快捷。

从以上可以看出，Moblin 的丰富图形界面是 Android 无法比拟的，因此 Moblin 得到了许多老牌的 Linux 开发企业支持，在 Moblin 上定制出自己的移动操作系统，目前市面上的常见的终端产品有爱国者的 MID 系列。当然 Android 也有类似的案例，如中国移动和其他几家运营商合作开发的 Open Mobile System（OMS）操作系统。

除此之外，Moblin v2 提供了比较完整的全套开发工具，并且在对程序语言的支持和执行效率上，比采用 Java 虚拟机的 Android 来的要好一些。不过目前 Moblin 在终端产品数量和厂商支持上和 Android 相比有一定的差距，我们几乎每天都可以听到许多厂商要支持 Android 手机或者介入 Android 上网本的制造的消息。

“开放手机联盟（一个由 30 多家科技公司和手机公司组成的团体）已开发出 Android，Android 是第一个完整、开放、免费的手机平台。为帮助开发人员开始开发新的应用程序，我们提供 Android 软件开发套件。”

同样，本段我们用一段官方表述开头。从这句官方表述中，我们很容易得知，Android 是一个

手机平台，作为目前市场占有率最高的移动终端设备，手机已经成为人们必不可少的随身物品之一。Android 正式瞄准这样的市场来进行研发。Android 的优势在于，合作伙伴够强势，品牌效应非常好，比起之前 Linux 系统在手机平台上得到的支持可谓天差地别，并且是一个开源的操作系统，发布近 1 年已经有大量成熟的应用程序支持，是中小型厂商所期待的产品，也是其他行业领头羊丰富产品线，实现深度定制，提高 ARPU 的利器。

Android 在最近正式推出了 Android SDK 1.5 版本，除了加入新的特性之外，还解决了许多 1.x 版本中的一些问题，如多语言的支持，输入法框架的支持，系统稳定性和操作界面 UI 等等。如果不是有 Google 的金字招牌，这样的产品换做别人发布可能很快会失去市场。

已经不止第一次有人对笔者抱怨说 Android 的操作界面和方式都不够友好。Google 更多的是以针对开发者的角度来设计这款操作系统的基础，让各类开发人员去完善这款操作系统，并且实现个性化，这在 Chrome 等其他 Google 产品的开发中也有所体现，也就是所谓的 Google Beta 精神，这种思路和策略与 Google 的企业文化息息相关。因此业内也有人将 HTC Dream 戏称为 Google Phone Beta。

Android 采用了 Java 作为应用程序设计语言，而底层采用了成熟稳定的 Linux 内核，应用程序依赖 Java 虚拟机来运行。这样的好处是很多现成的 J2me 应用程序可以很快被移植到 Android 操作系统上。同样，开放式的操作系统更有利于开发者开发优秀的应用程序，从而刺激软件消费。

不过 Android 目前的缺点同样明显，做为一款手机操作系统，他甚至不支持皮肤更换服务，这在以个性化为卖点的手机市场几乎是一个致命的缺点，另外采用 Java 虚拟机的 Android 对硬件系统也有一定的要求，无形中提升了产品门槛，因此 Android 并非大家所想象的是一款山寨型操作系统，对设备性能和规格都是有一定的要求的，普通的山寨 MTK 手机所采用的芯片组也不支持 Android 操作系统。

以上是针对 Android 和 Intel 的一些初步分析，两大操作系统各有优点，厂商和开发者可以针对自己所擅长的领域来决定采用哪款操作系统。

红帽 EMEA 合作伙伴峰会结束 参会人数上升 40%

近日消息，据国外媒体报道，开源巨头红帽公司日前表示，其最近主持召开的 EMEA(欧洲、中东和非洲)合作伙伴首脑会议取得圆满成功，此次参会人数创下了新的记录。

为期 3 天的会议在马耳他举行，提供给与会代表关于产品和战略的新理念和新资讯，以及宝贵的技术研讨和知识共享的机会。红帽透露，与会者来自 EMEA 地区的 38 个国家，与 2008 年相比，今年的参会者人数上升了近 40%。

会议所涉及到包括红帽虚拟技术研发状况在内的众多议题，红帽 CEO 吉姆怀特赫斯 (JimWhitehurst) 向合作伙伴发表他的了开源机遇的演说，分享他对于当前形势下开源模式的远见卓识。

红帽 EMEA 合作伙伴联盟主任佩特拉海因里希 (PetraHeinrich) 表示，开源的趋势和红帽的发展势头在合作伙伴生态系统中的优势越来越明显，因此该会议是红帽合作伙伴相互交流和增进了解的大好机会。

据悉，红帽将会在明年春季继续举办下一届 EMEA 合作伙伴峰会。

Linux 基金会:这一次微软是我们同床异梦的战友

Linux 基金会与微软就一项软件厂商需保证其产品没有质量瑕疵的提案联名上书, 提出了反对意见, 这一对死敌总算少见地取得了共识。这两家组织在写给美国法学院的一封信中称这项提案弊大于利。



微软的辩护律师 Horacio Gutierrez 在自己的博客中说: “微软与 Linux 基金会希望不要对现有法令作出更改, 这项提案将损害到商务与消费级软件市场, 也将对软件开发者们不利。”

Linux 基金会执行董事 Jim Zemlin 先生则在自己的博客上称他们与微软只不过是“同床异梦”的战友。不过他也认为这项提案将对开源软件组织和商业软件商造成伤害: “这种提案将打乱开源组织和商业软件商的正常操作模式, 并且将把他们带进不必要的官司漩涡中。”

“Linux 基金会和微软的这次合作看起来似乎意义非凡, 虽然我们的方针各有不同, 且我们发展的方针政策都不一样, 但为了软件业界的未来, 这次我们走在了一起。”

有趣的是, 就在这封联名信写成的几个月前, 微软刚刚控告一家公司在 Linux 内核中应用了他们的专利技术, 这家公司最后与他们达成了和解, 不过这预示着微软将来将采取法律手段限制 Linux 厂商在自己的产品中非法使用微软的专利技术。

对此 Gutierrez 也指出, 微软和 Linux 基金会此次的联合并不意味着对所有事情都持有相同的看法, “Linux 基金会和微软的通力合作是非常不寻常的, 尽管我们的差异吸引了更多的公众关注。”另外微软也向 Linux 基金会伸出了橄榄枝, Gutierrez 表示: “我希望这只是我们和 Linux 基金会众多合作机会中的一个, 这种合作今后还会更多, 合作可以带来更多。”

非洲上演开源之战

去年《华尔街日报》透露, 微软为了抢占非洲操作系统市场, 采用了挑衅性的商业营销战略, 比如用金钱诱惑学校从 Linux 转移到 Windows。

现在 BBC 报道, 微软非洲分部负责人 Cheikh Modibo Diarra 博士为微软的政策辩护, 称非洲是最后一个未开垦的疆域, 它的环境可以用一个词概括: 初级 (access), 非洲地区的软件、硬件和网络连接费用是美国的 50 到 100 倍。Diarra 博士说购买了微软的产品, 用户就能一劳永逸, 微软给用户的价格就是全部的费用。但另一方面, 非洲开源软件公司 Kiwanja 的 Ken Banks 指出, 在南非,

加纳，肯尼亚，尼日利亚和其它非洲国家，开源社区和开源程序员正茁壮成长。他认为用户在开源软件上的投入被夸大了，“不是所有的开源软件是难以使用和需要培训的”，他认为如果能接触源代码的话，将能更好的利用新出现的工具。

联通宣布加入开放手机联盟 将开发手机操作平台

5月18日消息，中国联通今日宣布将加入 Google（谷歌）公司组建的开放手机联盟（OHA），这意味着中国联通将在市场上开发支持 Android 平台的手机和移动服务。业内人士预计，这一举措将对中国联通 3G 手机产业链和未来用户体验产生重要影响。

中国联通与手机开放联盟的合作将是双方 3G 发展战略的一部分，中国联通将与手机开发联盟合作，在中国市场推出支持 Android 平台的手机设备和基于 Android 平台开发的移动服务，其中包括基于 Android 平台开发的手机输入法、语音搜索等服务。此外，中国联通和谷歌将在联合推出的 Android 手机上进行移动搜索的合作。



中国联通表示，将与开放手机联盟的其他成员一起，共同促进和开发更为丰富多样、更低成本和更好体验的移动产品。中国联通的加盟使得开放手机联盟的全球成员达到 48 个。

“中国联通非常高兴加入手机开放联盟，我们期望为联通用户提供更多的基于开放的 Android 平台的产品和服务。”中国联通副总裁李刚表示，“手机开放联盟的合作方式有助于发挥无线技术的潜力，它帮助我们在移动通信领域建立新的协作环境。同时，Android 是目前开放性最强的移动设备综合平台，它使得电信运营商、手机制造商以及产业链各环节能够更好、更快地将创新性产品推向市场，同时能有效的降低成本。中国联通将在中国市场推出支持 Android 平台的手机设备和基于 Android 平台开发的移动服务，更多的中国联通用户将因此受益。”

谷歌全球副总裁、大中华区总裁李开复表示：“谷歌公司将积极与中国联通开展合作，与手机开放联盟的其他合作伙伴一起，合作开发更为实用和多样的基于 Android 平台的手机软件，谷歌的搜索服务也将优化移植到手机平台，我们期待更多的中国用户能够受益于价格更低廉的移动产品，更优质的移动服务，更丰富的移动互联网应用，以及更易使用的手机界面。”

Android 是 google 开发的基于 Linux 平台的开源手机操作系统。它包括操作系统、用户界面和应用程序移动电话工作所需的全部软件，而且不存在任何以往阻碍移动产业创新的专有权障碍。这使得全球运营商、手机商纷纷加入。

开放手机联盟是一个跨国界联盟，致力于培养移动产品创新，并为用户提供创新性的移动体验。目前手机开放联盟已经加盟的成员是来自全球各个国家的手机制造商、软件开发商、半导体公司和其他商业企业，其中包括西班牙电信、沃达丰电信、东芝、爱立信、华为等在各自领域具有巨大影响力的企业。

十个优秀的特征 让你喜欢上 Android 1.5

我们期待的 Android 终于在 4 月底发布了最新版本，我们来深入的看一看十个最优秀的服务用户的特征吧。

上个月，Google 官方已经发布了 Android1.5 更新，取名“cupcake”。Geek 的 JoelEvans 有机会简短的接触这个 beta 版本，也足够给我们第一印象了。仅仅一个月过后，这款新软件已经显然的可以安装在 Android 驱动的设备上了。别搞错，Android1.5 是一个重要的升级。这款软件带来了许多新的能力，其中的一些是在其他的平台上面看不到的。



Android 1.5 虚拟键盘，桌面 widgets，Google 语音搜索，其他的好用的功能。

这是个重要的更新！

首先，这个更新的操作系统运行在 Linux kernel 2.6.27 上面，这意味着更强大的安全性，增强的稳定性和一系列的核心应用加强。Android 1.5 现在携带 SIM Application Toolkit 1.0 和自动检测特征，以及 SD 文件系统修复功能。就像 iPhone OS 3.0，Android 1.5 携带 SDK 1.5，加入新的 API，可以帮助开发者创建更好的应用。更重要的是，为终端客户的这些新的特征提高了使用感受，十分有意义。

其中的一些特征轻松的赶上了 iPhone，比如新的虚拟键盘和提高的手机网页浏览器。其他通过华丽的外表被设计加入更多的实质，比如栩栩如生的、很顺滑的窗口变化，加速度模式程序，使照片和风景模式切换，以及优美的用户界面元素。

更新给予 Android 更多的魅力

Android 1.5 也带来了很强的性能，甚至超过了 iPhone OS 3.0，至少在某些方面，比如语音搜索，桌面 widget 和 Live 文件夹。

所有的这些加在一起，Android 1.5 应该可以真正的宣告 Android 2.0 根据新颖特性的数量。也证明 Google 致力于完美 Android 平台，没有任何意愿让 Apple 的 iPhone OS 3.0 更出风头。当带

给你更多的简单、效率时，Android 1.5 的这些进步保证该产品具有更强的竞争力。

“cupcake”在 5 月 11 日到来

虽然 Google 没有设定 Android 1.5 固定的发布时间，搜索巨头表示该操作系统依据特定市场和个人选择部署更新的时间，并不像 iPhone 固件更新，被下载和安装通过桌面 iTunes 软件。Android 1.5 将通过无线网络更新的方式。

你的 Android 手机将提示你下载安装更新，当该更新有效时。就像先前的声明，T-Mobile 预计开始部署更新在美国的 AndroidG1 上，就在这个周一。推出将分步骤，所以不要焦虑如果你还没有马上得到。就像我们期待这个更新，我们选择了 10 个新的特征，强制安装在每个 Android 平台的 1.5 的特征。

1. 智能虚拟键盘

虚拟键盘的增加意味着 Android 1.5 设备可以同时支持物理和虚拟的键盘。不同的输入方式的选择满足用户特定场景的使用需求。Android 虚拟键盘可以在任何的应用中被提供，包括 Gmail、浏览器、SMS，当然也包括第三方应用。

它包括自动校正、推荐、习惯文字的用户词典。你可以通过振动屏幕来进行触觉反馈。不同于竞争对手的手机平台，Android 1.5 支持第三方虚拟键盘应用的安装。

2. 使用 widgets 实现桌面个性化

你可以用 widgets 武装自己的桌面，基本上小的 web 应用从网络调用和展示实时数据。Android 预装 5 个打包的桌面 widgets，包括数字时钟、日历、音乐播放器、相框和搜索。不同于 iPhone，Android 1.5 通过内置的应用程序库安装第三方 widgets。

3. 在线文件夹(Live Folder)快速浏览在线数据

类似 OS X Leopard 的 QuickLook 特征，Android 的在线文件夹显示常见的数据条目，比如联系人、喜欢的应用，email 信息，播放列表，书签，RSS 种子等，而并不需要运行系统程序处理特定数据条目。

在线文件夹数据更新实时，就像通过云或是本地创建新的数据。什么是最好的，开发者可以拓展通用数据条目和注册新数据类型的内置支持。

举例，Twitter 客户程序可以注册 tweet 作为新数据类型，因此可以使你从你的朋友那里创建 tweet 的在线文件。Android 1.5 带来一组在线文件夹提供到在个人的桌面，来提供快速、方便的进入你的联系人、股市、书签等。

4. 视频录制和分享

Android 1.5 最终加入了录制、分享视频的功能，另外增强的回放和 MPEG-4、3GP 视频格式的更好的支持。你可以通过 email、MMS 或是直接上传到 YouTube 来分享录制的视频，使用隐私控制来在选择是分享给朋友还是每个人。上传的同时，你可以继续使用手机，甚至在上一个上传继续时，还可以继续录制和上传新的视频。

5. 图片上传

在线分享图片需要更少的点击。完成照相后，当浏览图片或是选择 Google 在线图片服务” Picasa “时，只需轻点” 分享 “，并会带给你 1G 的免费图片存储空间。

6. 更快的标准兼容浏览器

Android 的基于 Webkit 内核的浏览器带来了重要的调速装置 (Speedpumb)，感谢新的 Webkit 渲染引擎和优化的 Java 脚本编译器” SquirrelFish “。结果是，你可以享受更快的性能表现，当遇到重度使用 java 脚本代码的复杂的 web 应用时。留心的用户可以注意到后一个也支持当前的 Apple 桌面和 Safari 浏览器。Desktop Safari4 Beta 和 iPhone OS 3.0 的手机 Safari 运行一个新的和更优化的版本 SquirrelFish 增强版。

除提高速度外，Android 的浏览器支持 web 页面内的复制和粘贴，用户可选择的文本编码，统一的前进搜索框和页面内的文本搜索能力。

7.Voice search 语音搜索

iPhone 用户接触带有语音识别技术的 Google 手机搜索在去年的 11 月份，尽管如此，在 Android 1.5 中包括了语音搜索特征的确是增强了默认搜索能力，超过纯文本搜索。

这个进步可以满足那些选择无论是英语还是美国口音说出疑问的人。语音搜索工程进展的不错。当你大声说出你的问题的时候，软件将上传数字信号记录到 Google 服务器中。在那里，艺术级别的语音识别技术转化语音到特定的文本搜索，使之通过 Google 搜索引擎。通过地理位置的赛选，将给过反馈到手机设备。

8.立体声蓝牙和免提电话

除了增强的免提电话体验，Android .1.5 支持立体声蓝牙 (A2DP 和 AVCRP)，还有自动配对功能。

9. 时尚的整体性能

代码库的优化渗透到 Android 核心应用中，使感觉时尚。预期更快的摄像启动，图片捕捉，除了浏览器顺滑的浏览器页面和 Gmail 对话列表滚动之外，软件请求你的 GPS 地址显著加快，这要感谢 SUPL 支持的 GPS 技术。

10. “锦上添花”

Android 自动检测和修复 SD 卡的文件系统，允许第三方应用显示 Android 系统硬件特征，为了更合适的性能。Android 当下载应用时检测这些需求，只在所需求硬件存在的情况下才允许安装。当更多的 Android 设备建立在不同的硬件上时，这个功能会显得很实用。

Gmail 客户端为了巨大的存档文件有个新的功能，立即删除或是标注执行在多个信息上。Google Talk 朋友状态在联系人、MMS、GMail 和 Email 应用中可见。你可以显示收藏联系人头像，一键从呼叫记录中进入联系人名片。联系人记录事件显示日期和时间标记。最后值得一提的是，使用在跨系统程序和第三方应用程序的用户界面元素被提炼出来，为了更好的精致的外形界面设计。

Novell 张先民：Linux 发展中心渐转向中国

近日，Novell 公司数据中心创新大会在京举行。会上 Novell 向来宾介绍了发布不久的全新 SUSE Linux Enterprise 11 的全新性能，充分展示了 2009 年业界发布的重要平台，及操作性最强的 Linux 解决方案。同时，Novell 还在本次数据中心创新大会上展示了最新的数据中心管理解决方案。

大会现场，Novell 东亚区总裁张先民先生在接受记者采访时表示，SUSE Linux Enterprise 11 发布到现在，已经有了一百万次的浏览量以及五十万次的实际下载，下载用户中相当数量下载来自于

中国，其中 50%左右的下载来自于社区，50%来自于客户和集成商。在谈到 Linux 的发展变迁时，张先民认为，过去十年实际上一一直是国外主导 Linux 发展，但从上网本时代开始，随着 Linux 本身易用性便捷性的提高，以及在中国逐渐受到越来越多用户的青睐，中国将成为推动 Linux 应用发展的中心之一。

上网本将推动 Linux 发展

具体在上网本的应用上，Novell 东亚区总裁张先民介绍，Novell 通过 Moblin 2.0 项目上，与英特尔展开了广泛的合作。Moblin 是一个由英特尔领导的，Linux 基金会扶植，以硬件为导向的操作环境，Novell 作为设计合伙人之一，在其中做出了卓越的贡献。包括：ANJAL 邮件客户端，Cubano 多媒体播放器，改进网络堆栈应用等多个方面。Novell 和英特尔在五月发表共同新闻声明，Novell 将协助英特尔加速 OEM 以及 ODM 采用 Moblin 2.0，并于台湾成立 Open Labs，与英特尔位于台湾的 MEC(Moblin Enabling Center)共同合作进行 Moblin 相关的合作开发。



另外，Novell 高级副总裁兼开放平台解决方案事业部总经理 Markus Rex 也从 Novell 在普及性、互操作性、关键任务和物理与虚拟计算三大主题的创新和实践，现场做了详细阐述。

在谈到 Linux 的普及性时，Markus 认为当前 Linux 已经无处不在，且已经实现广泛支持各类新技术，包括云计算、虚拟环境，真正做到了在任何环境、任何地点以任何方式实现。

在谈到互操作性问题时，Markus 认为 Novell 为用户建立了桥梁，将原来互不沟通的世界打通了，到现在数据中心采用混源环境已经成为一种标准，这对客户极具吸引力。从 01

年就开始启动的 MoonLight 项目也取得了丰硕成果，让 .NET 开发人员可以在不同的平台上执行他们的应用程序，涵盖 Linux、Mac OS 及 Unix，同时造福广大的开发人员、独立软件开发商及使用者，让他们只要撰写一次软件就能在不同的平台上执行。从用户角度，过去很多丰富的英特网资源只能被 Windows 的用户通过 Silverlight 项目来享用，现在同样可以被 Linux 的用户所享用了。在实施的方面则完全实现了开放源代码。

Novell 在中国的增长

谈到 Novell 在中国的增长，张先民博士特别提到的是大客户和 GSP OEM 合作的突破。在中国的华为和中兴，自制刀片服务器已经得到大规模的推广，而这些服务器操作系统则大量采用 Novell Linux Enterprise。华为已经在大规模部署此类安装有 SuSE Linux 的刀片系统；而中兴目前也在做，目前已经开始测试，3-6 个月内将会大量部署。另外，Novell 和国内其他服务器厂商的合作也没有停止，比如 Dell、HP 和宝德，每年都保持在两位数的增长。在和 IBM 的合作方面，目前主要是在大型机 Z 系列上，特别是目前政府主导的政府信息工程绿色化的指导意见下，在整合后的 Z 系列服务器上，跑的都是 SUSE Linux 操作系统。

社区扫描

Cisco 将会召集到下一代 Linux 开发人员？

传统观念认为，Linux 应用程序开发人员往往是围绕在发行版厂商周围的，比如红帽、novell 和 Canonical (Ubuntu 母公司)。但是现在来自 Cisco 的一项举措可能使得目前的格局有所改变。因为从去年开始，Cisco 为他的统一通信产品举行了一个基于开源的软件设计大赛，主要的目标就是吸引 Linux 平台的开源软件开发人员。



这个名为“应用扩展平台开发者大赛”的比赛是由 Cisco 发起的，从去年 10 月份开始，Cisco 的 AXP 是一个基于 Linux 的开发平台，而这个 AXP 平台正是 Cisco 的集成服务路由设备的软件基础。所以说这个开发平台很重要，因为它是 Cisco 和微软的统一通信大战的一个重要砝码。Cisco 推广他的统一通信，是基于网络设备，而微软的统一通信（微软自己称统一沟通），作为统一通信整体的解决方案，cisco 和微软谁也离不开谁，但是 Cisco 在尝试用基于 Linux 的软件的来构建他的统一通信解决方案。

自从去年 10 月份开始，这个活动就在 Linux 开发者社区引起了不小的轰动，到目前为止，cisco 已经收到来自 75 个国家和地区，超过 900 个的参赛作品，形式非常喜人。目前 cisco 已经宣布了第一阶段获得最终大奖提名的 10 位开发者名单。第二阶段即将开始。Cisco 将继续和微软较劲到底？

MySQL 创始人宣布创建开源数据库联盟

MySQL 创始人 Monty Widenius 以及另一个 MySQL 专家于 5 月 13 日宣布创建开源数据库联盟 (Open Database Alliance)，以便应付甲骨文收购 Sun 之后对于开源数据库的不确定影响。

Monty Widenius 在声明中说道，创建此联盟的目的在于“为 MySQL 的发展提供一个稳定的空间，以便推动一个真正的基于社区参与的开源开发环境，同时保证 MySQL 的代码拥有极度优秀的质量。”他还表示此联盟会“与任何使用 MySQL 的企业展开合作”，甚至包括 Oracle 以及 Oracle 数据库开发人员。

OpenOffice 前途惨淡，我们还有其它选择吗？

随着甲骨文收购 SUN，OpenOffice 的前景让粉丝们都捏了一把汗。也随着开源的大行其道，更多人开始关注开源 office 软件，这次就来盘点下开源 office 软件。

OpenOffice.org 是一套跨平台的办公室软件套件，能在 Windows、Linux、MacOS X (X11)、和 Solaris 等操作系统上执行。它与各个主要的办公室软件套件兼容。OpenOffice.org 是自

由软件，任何人都可以免费下载、使用、及推广它。

OpenOffice.org 现在的最新版本是 OpenOffice.org3。OpenOffice.org3 是一个逐渐进化的而不是革命性的产品，它是 OpenOffice.org2 升级的产物。与之前的版本相比，OpenOffice.org3 并没有进行彻底的改动。

OpenOffice.org3 的最大变化实际上是很多小变化的集合。举例来说，Mac 用户很乐意知道 OpenOffice.org3 现在已经是本地应用程序了。该套件中的所有的应用都可以通过一个简单的统一界面或者系统托盘启动器启动，后者同时也可以作为该套件的加速器。OpenOffice.org3 的整性能明显要好于它的上一代版本，即使是不使用应用程序启动加速器也是如此。

OpenOffice.org3 新增的其它功能包括方程求解器和计算器工作簿共享功能；Impress 支持多个监控器；支持 VBA 宏。

你可以使用 OpenOffice.org3 直接打开 Office2007 的 OOXML 格式文件，但是你却不能将文件保存为 OOXML 格式，因为已有的 OOXML 文件中的某些东西似乎不能实现正确转化。比如，如果某个文件在同一行包含多种言语的字符(比如英语和日语)，那么在导入时它的行距就会遭到破坏。不过，如果你将该文件保存为 Word2003 文件格式并导入就不会有任何问题。



OpenOffice 的一个我一直非常喜欢得功能就是能够直接输入 PDF 格式，而不需要一个插件或虚拟打印机驱动程序，同时还具备加密和表格功能。在 OpenOffice.org3 中，PDF 输入器的选择范围稍微有些扩大；最大的升级就是支持 PDF/A。

不过，OpenOffice.org3 的一些小问题仍然让我很苦恼。例如：文件窗口底部的各种工具不是那么容易让人理解---你也许永远不会知道标有 STD 的小方块或者它旁边只标有星号的小方块是干什么用的，除非你参阅使用手册。这一点其实非常不方便，当你开发一个供普通用户使用的软件时，易用性非常重要，使用手册是最后才被用到的手段。

记住，OpenOffice.org 并不是唯一的。GoOO 与 OpenOffice.org 使用的源代码库相同，但是它在速度上面进行了一些优化。在日常工作中，我一般交替使用 GoOO 和 OpenOffice.org，因为二者的文件格式完全兼容，而且前者的速度和稳定性都稍快。PortableApps 版本的 OpenOffice.org 可以在可移动驱动器上运行，因而可以用作该软件的一种试用方式，省去了正式

安装带来的一些问题。

IBM Lotus Symphony

IBM Lotus Symphony 是一组集文字处理、电子表格和演示幻灯片于一身的免费办公套件。它支持开放文档格式（Open Document Format，简称 ODF）。ODF 最大的优势在于其可开放性和可继承性，基于 ODF 格式的文档在许多年以后，仍然可以为最新版的任意一款遵循标准的办公软件打开使用。与此同时，IBM Lotus Symphony 兼容 Microsoft Office 格式。具备丰富功能的生产力工具，直观的用户界面，并能与各种文档兼容。无论在家里、学校或公司都可以免费使用。

虽然也是基于 OpenOffice.org 的代码库，但是 IBM Lotus Symphony 却截然不同。最近推出的 IBM Lotus Symphony 使得濒临死亡的 Lotus Symphony 品牌重新焕发了活力。它包含的功能要比 OpenOffice.org 少得多---该套件之包含三个主要的应用软件---但其中的每个软件都值得我们关注。

IBM Lotus Symphony 的三个应用软件正好满足了大多数人使用办公套件所要完成的工作：写文件、创建电子表格并组装演示文稿。但是，这并不意味着该套件在以后不会增加其它新的功能，并且它能够自动检查新的组件和对于现有软件的升级。IBM Lotus Symphony 不支持 OOXML，因此不能实现与 Office2007 的转换，目前似乎也没有一个插件能够胜任这项工作。

使用 KOffice 套件，用户能够在一个标签式界面上运行所有的应用程序。IBM 替换了 OpenOffice.org 默认的图标和工具，取而代之的是一套更好、更精致的元素集，其中包括一套用于文本属性的可停靠侧边面板，这在原始程序中是不具备的。也就是说，每一个应用中的功能组合与 OpenOffice 本身几乎是完全一样的。

很显然，IBM 想要将 Symphony 变成一个后续开发的新起点和框架，无论是于 OpenOffice 保持同步还是分离。从 Symphony 的插件集上你就能明白这一点，虽然它效仿了 OpenOffice.org 的插件集，但是这些插件是完全不同的，因为 OpenOffice.org 的插件不能用于 Symphony，反之亦然。虽然 Symphony 目前没有太多可用的插件，不过仅有的一些全部都是精品并且非常有用，比如有个插件能够将演示文稿输出为单独的 Flash 文件。

StarOffice

StarOffice 是太阳微系统公司（Sun Microsystems）所属的专有办公软件套件。它原先由 StarDivision 公司开发，1999 年 8 月被 Sun 收购。该软件套件的源代码于 2007 年 7 月释出，衍生出一个自由开放的开源办公软件，叫做 OpenOffice.org；StarOffice 的后继版本皆基于 OpenOffice.org，并带有另外的专有组件。

StarOffice 和原始的 OpenOffice.org 从外面看几乎没有任何差别。启动之后，除了一些有关商标的改变外，你所看到的界面是完全相同的---甚至在经过一些深入得比较之后，你会发现二者的功能集也是相同的。那么，你花钱来购买 StarOffice 的意义何在？

答案包括三个方面：支持、捆绑和部署。正如 Canonical 向 Ubuntu 用户提供付费支持一样，Sun 也对 StarOffice 提供同样的技术支持。具体的支持项目则要视你所使用的版本而定。下载版和标准版都配有最长为 60 天的技术支持；而按照每台电脑进行授权并配备额外部署工具的企业版，使用的则是一个单独的支持合同。

在应用程序捆绑方面，StarOffice 预装了开源的 Thunderbird 邮件客户端和 Lightning 日程表，如果您选择购买标准或企业版。而使用下载版本的客户则需要手动添加这些组件，这与原始的 OpenOffice.org 在下在和配置上没有任何本质上的差别。

此外，企业版的 StarOffice 还包含一些其它版本不具备的功能，其中就包括微软 Office 宏转换工具(OpenOffice.org 可以让你按原样运行 VBA 宏，但却不能将它们转换为 OpenOffice.org 本身的宏语言)和 NetBeans 扩展套件。

Sun 公司对 OpenOffice.org 生态系统做出的重大贡献，除了 StarOffice 本身以外，那就是它为该系统开发了众多的实用插件。PDF 导入和编辑扩展对于基本的文字或格式调整非常有用，但对于更加高级的处理却不是非常适用。有些 PDF 文件---其中包括用 OpenOffice 本身生成的文件---不能实现导入。博客发布扩展能够与所有主要的博客 API---WordPress、Movable Type 和 Metaweblog 兼容，但是它提供的功能与专业的博客工具相比，可以说是微不足道的。

就我个人而言，我特别喜欢 MediaWiki 扩展，它可让你在使用 MediaWiki 软件的网站(比如 Wikipedia)上直接进行编辑和发布，而无需通晓 MediaWiki 标记语言。

Koffice

文本编辑器 KOffice 是一组办公室套装软件适于用 K 桌面环境 (KDE) 环境之上，所有的套件都是采用开放源代码架构。KOffice 最显著的特定之一就是所包含的应用软件范围最广，种类最多。KOffice2 的第三个测试版已经被证明稳定性非常好，不过仅限于 Linux 操作系统。不过，非官方的 Macintosh 版 KOffice2 已经出现，并且有一些项目正在研究将 KDE 和 KDE 兼容的应用程序引入 Windows 环境中，但是到目前为止，能够在 Windows 环境下使用的唯一一个 KOffice 应用就是数据库应用 Kexi。

为了得到最大的兼容性，我在 Kubuntu 下尝试了 KOffice 下，Kubuntu 软件包中自带 KOffice。并非所有的 Linux 发行版都包含 KOffice。请注意，虽然 KOffice 在 Gnome 或 KDE 下都能运行，但是如果你要得到最好的体验最好使用后者。每一个单独的 KOffice 应用都可以独立运行，或者在 KOffice Workspace 框架之内，所有的应用以一个标签式的界面运行。

除了包含文字处理、电子表格、演示文稿和数据库系统之外，KOffice 还包含三个图形处理工具---流程图应用、矢量绘图软件和光栅设计应用---项目管理工具、报表开发工具、制图绘图工具和数学公式编辑器。这对于那些需要使用这些功能的用户来说，KOffice 是一个很好的选择，不过前提是他们要具备运行 KOffice 的平台。

广泛的工具集并不是 KOffice 的唯一优点。这些应用本身在展示和布局方面汇集了开发人员相当多的创意---它们在导航方面非常容易上手，而且 Krita 中的面板以及图形编辑软件在布局方面是一致的。Krita 有一件事情一直困扰我：它支持 CMYK 色彩空间，但实际上却没有 CMYK 模式的颜色选择器。

此外，KOffice 在跨格式上还存在一些问题，不过我觉得这很可能是因为试用版原因。我在 Word 2003 种创建了一个文件，然后用 OpenOffice 将该文件保存为 ODF 格式，最后用 KOffice 打开该文件，发向有一些格式发生了错位。当我用 KOffice 重新打开原始的 Word 2003 文件时，没有出现任何问题。并且，通过 KOffice 或 openOffice 创建的文件在保存和打开时也没有出现问题。

AbiWord

上述开源办公软件各个都是名声响当当，不过 AbiWord 也确实值得一提。abiword 是一个自由的文字处理程式类似微软文字处理软件---Microsoft Word，它适用于各种各样的文字处理任务，由广大的开源社区用户进行后续开发和维护。AbiWord 的功能库没有 OpenOffice.org 那样广泛---它只是一个字处理器，而不是一个功能完整的应用套件。不过，AbiWord 却引起了很多人的关注和喜爱，因为它仅仅占用 15MB 左右的硬盘空间，并且运行速度惊人。

顾名思义，AbiWord 基本上是经典的 Microsoft Word 核心功能的克隆。你平常所使用的几乎所有的 Word 功能---封、报表、邮件合并、修改跟踪---都可以在这里找到。AbiWord 并不适合企业级专业人员使用，但是对于那些工作要求不是那么严格的学生和个人来说，AbiWord 足够了。AbiWord 的另一个优点就是它也可以跨平台，所以如果你刚开始的时候是在 Windows 上使用它，那么在 Linux 上你也完全能够使用。

经典的 Word 文档都可以用 AbiWord 打开，而且 AbiWord 也具备打开 OpenOffice 和 ODF 格式文件的插件，不过对于它目前尚不支持 Office2007 OOXML。但是，并不是一切都非常完美，例如，当我试图打开一个包含多种语言（英文和日语）时，非英文文本没有正确显示。鉴于此，AbiWord 最好用于创建全新的文件，而不是处理原有文件。

此外，AbiWord 的一些小功能非常招人喜欢。比如，它的 HTML 导出功能可以让你使用 base64 编码封装任何图像文件包括 HTML 文件本身。AbiWord 第三方插件功能丰富，大大扩展了该软件的功能。最后，你可以在 PortableApps 包中运行 AbiWord。

Google Docs

谷歌办公套件，类似于微软的 Office 的一套在线办公软件，可以处理和搜索文档、表格、幻灯片，并可以通过网络与他人分享，有 google 的帐号就能使用。

Google 文件是基于网络的文字处理和电子表格程序，可提高协作效率。有了此程序，您就可以把文档放在网上。这样，您团队中的其他人就可以实时在线对文档进行编辑和更新，免去您线下发送相同版本的文档并根据每个不同的回复对相同文档进行反复更新。多名用户可同时在线更改文件，并可以实时看到其他成员所作的编辑。系统会自动为您保存每次的修订，这样您就可以看出来谁在何时作了什么修改，并可恢复到之前任何一个时间点的旧版本。

文件可以很方便地从 Google 文件导入及导出。若要从保存在您计算机上的现有文件开始操作，只要上传该文档，并从您上次停止的地方继续进行即可。要离线使用文档或将其作为附件分发，只需以最适合您的格式在您自己的计算机上保存一份 Google 文件副本即可。无论您是上传还是下载文件，所有的格式都会予以保留。

就像 Google 的其他网络服务一样，您不需要下载或安装另外的软件。您只需一台接入互联网的计算机及可使用 Google 文件的标准浏览器即可。

是 Apache 还是 GPL 更有利于开源商业？

SourceForge 网站托管的开源项目七成以上是采用 GPL 许可证。尽管如此，Eric, S.Raymond 坚持认为 GPL 是“恐惧和虚弱的告白”，由于 GPL 引起的不确定性和恐慌，降低了开源的普及和推广。Raymond 所持观点是如果开源是最佳的软件开发模式，那么使用最开放的许可证也许最有意义。

Apache 软件基金会的 Geir Magnusson 指出纯 GPL 项目不多，成功的开源项目很少只使用一种许可证。为了妥协 GPL 固有的和强制性的“自由”权利，开发者不得不采用某些绕过限制的方法。最近 Linus Torvalds 在电子邮件中评论 GPLv3，“GPL/GPLv3 anti-ASP/TiVo 条款保护不了任何事情，它无法阻止抄袭者。”开发者可能偏爱 GPL 风格的许可证，但对于企业而言约束较少的 Apache 许可证对他们的产品更有帮助。旧的 GPLv3.BSD/Apache 争论必须面对商业现实。

开源社区争论话题之三：使命何在？

酷爱争论让人们聚集在自由开源社区中，同样也因为这个原因，人们对自由及开源软件的使命有不同的观点。

对有的人来说，FOSS 只是意味着无需花钱就可以免费使用的免费软件；而有的人则认为源代码的开放性更为重要；而还有一部分人(以 GNU 创始人理查德·斯托曼(Richard Stallman)为代表)则认为，FOSS 的首字母“F”才是自由及开源软件的使命所在，即自由应超越技术、超越政治、超越地域。

尽管观点不同，但有一点无需怀疑，人们在捍卫各自观点的同时，自由及开源软件正在悄然向前发展。无论 FOSS 的使命可能是什么，它实际上正在很好的满足人们的多种需要。

FOSS 正在走向成功，这不仅仅体现在越来越多的人开始使用它。那么它是否在实现自己的使命呢？需要指出的是，这个问题的答案取决于你对其使命的看法。

包容异见是 FOSS 最大特点

Linux 内核创始人李纳斯·托沃兹(Linus Torvalds)表示，“我认为它的使命不只有一个。”

托沃兹解释道，“实际上，我喜欢开源的原因是，它可以让人们去完成不同的任务，从不同的角度去考虑问题，最终的结果并不一定与当局者最初的设想完全一致，但却一直在向更好的方向发展。”Linux 本身就是这样一个例子，托沃兹解释道，“当我最初创建 Linux 的时候，我没有远大的目标，如同目前的内核一样，今天所具有的所有发展动力来自于外部，来自具有不同需求及为这些需求而努力工作的人们。”

托沃兹将开源比作“一个更具成长性的开发模式”，他表示，“对我来说，它与‘生活’非常相似，有谁知道生活的最高意义何在？我不知道，我不认为有哪一点的重要性能够远胜其它。但我们依然会努力的生活和工作，分享我们的知识和收获，更好的改变我们的环境。”开源也一样，托沃兹表示：“开源与生活具有相似的一点，‘改善你的环境’并且‘分享你的知识。’”

FOSS 是一种‘爱心传递’

《精通 vi 和 Vim》一书的联名作者艾尔伯特·汉纳(Elbert Hannah)表示，使命问题难以有一个一致的答案，“10 个人可能有 10 个不同的答案，不过这些答案的中心意思一般都可以追溯到理查德·斯托曼的观点。”

汉纳表示，一般来说人们对 FOSS 的理解是，开发软件并将其源代码免费对外开放，“这是一种‘爱心传递’，如果你使用并修改自由软件，如果你把它作为一个产品发布，你必须也公开自己的源代码。”有的人认为，通过支持和服务可以带来收入，但是有的人则支持“免费的服务或支持”。

汉纳表示，自由软件已经获得了很好的发展，我们现在拥有一个免费的高品质 Linux，已经堪称一个奇迹。“在很多方面 FOSS 可以说已经获得成功，不过我不确定在市场和环境中，它能否继续生存下去。”“我对自由软件做出了回报，”他补充道，“但是我也需要谋生，对于某些我自己开发的应用工具，我不得不决定是否免费对外公布它们，我也希望可以出售它们中的一部分来获得生活所需的收入。”

自由与开源的区别

来自蒙特利尔的开源顾问格哈德·麦克(Gerhard Mack)表示，FOSS 的主要目的是“尽可能的

满足计算机用户的多种需要，我对它的成功充满信心，目前我很少使用闭源软件。”

Slashdot 博客马丁·埃斯皮诺萨(Martin Espinoza)指出，分清自由软件和开源软件之间的区别，是非常重要的。埃斯皮诺萨表示，“‘开源’只意味着人们可以看到源代码，这对于互操作性是一件好事情。另一方面，‘自由软件’强调软件必须保持基本的自由性，其中有一些是那些基于服务的软件发行模式所尽力避免的东西。”埃斯皮诺萨表示，所有自由软件都是开源的，但并非所有开源软件都是自由软件。“开源软件实质上是一种折衷的办法；它可以确保互操作性，”埃斯皮诺萨解释道，“如果你自信产品具有竞争优势，那么你必须让其具有互操作性。”

当然，并非所有人都认为这个话题具有争论的必要性。来自教育行业的博客作者罗伯特·普森(Robert Pogson)表示，“我没有看到自由及开源软件有什么区别。世界需要信息技术，需要软件。有很多天才在创建和分享软件。自由/开源软件是全球协作实现的软件。”他表示，“如果有人不喜欢自由/开源软件运作的模式，他们可以花费更大的代价来自己开发软件，共享是开发软件的更有效方式。”普森表示，自由/开源软件是非常自由的，它让人们可以灵活的解决任何问题。他认为，“自由/开源软件可以给人们带来巨大价值，人们无需再花费数十亿美元来购买软件许可，而是可以亲自去贡献/安装/修改它，可以成为一个充满活力环境中的一部分，而不仅仅是机械使用软件的客户。”

和托沃兹一样，博客凯文·迪安(Kevin Dean)认为，对自由和开源软件来说，没有什么最高使命，它不是一个单一的目标、目的或运动。他表示，“自由/开源软件和一个森林一样，其中的树木都按照自己的方式、自己的节奏在成长。”

FOSS 支持者的希望与梦想

如果自由和开源软件没有所谓的“最高使命”，那么如何来评价未来数年中它是否取得成功呢？或许只有看它是否实现了其支持者的希望，当然，会有很多不同的希望。托沃兹表示，“我只是希望继续推动 Linux 朝着有趣、有用的方向发展，目前这个目标正在实现。”

埃斯皮诺萨表示，他希望所有软件最终将变为自由免费获得，也就是说，用户无需花钱就可以使用软件。他表示，“商用软件总会有适合自己的一部分市场，不过我们正在发现，开源软件逐渐超越每一个闭源软件，因为它们在不断进步。”埃斯皮诺萨断言，尽管 Linux 不能解决世界上的所有问题，但是如果多一些开放性和自由，世界将变得更加美好。

汉纳表示，他希望自由软件能够在竞争激烈的技术市场占据一席之地，在这一方面，Linux 和 GNU 软件的表现一直不错，从而避免微软的一家独大格局。另外，自由软件还可以为服务于研发和技术提高。汉纳认为，自由软件是解决人们难题的一支关键力量。

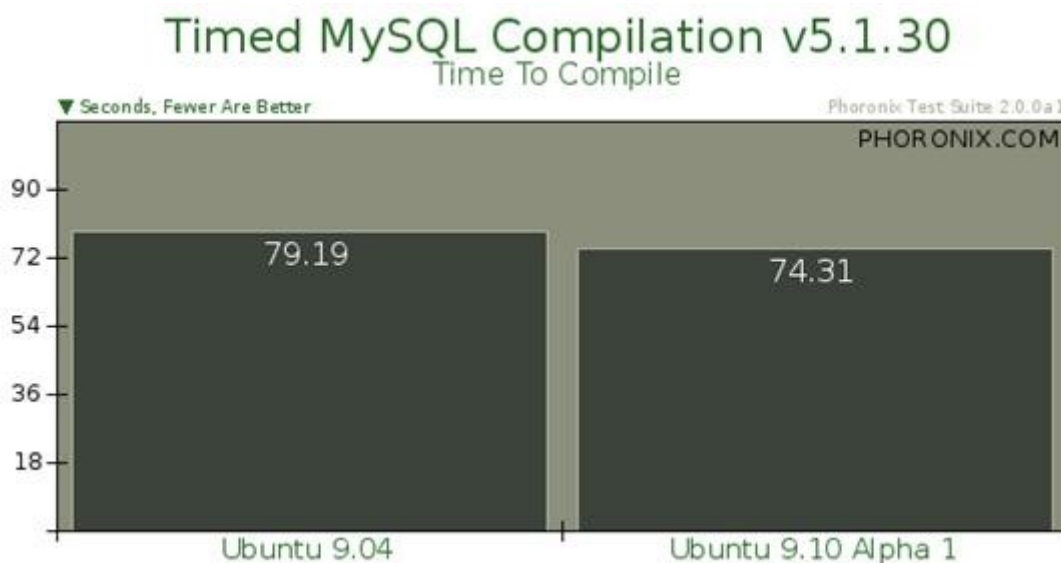
而迪安则表示，他希望 FOSS 支持者能够意识到，版权问题也是 FOSS 需要面对的一个问题。微软曾经挥舞着版权法律的武器，来强迫人们按照微软的方式来使用程序；如果 GNU 或自由软件基金会也使用版权法来做类似的事情，同样令人感到厌恶。

不过，无论事情如何发展，无论技术如何发展，有一点可以肯定：总会有人热爱和支持自由和开源软件。普森表示，自由和开源软件是一个宝石，值得人们去赞扬、雕琢和自豪的使用。

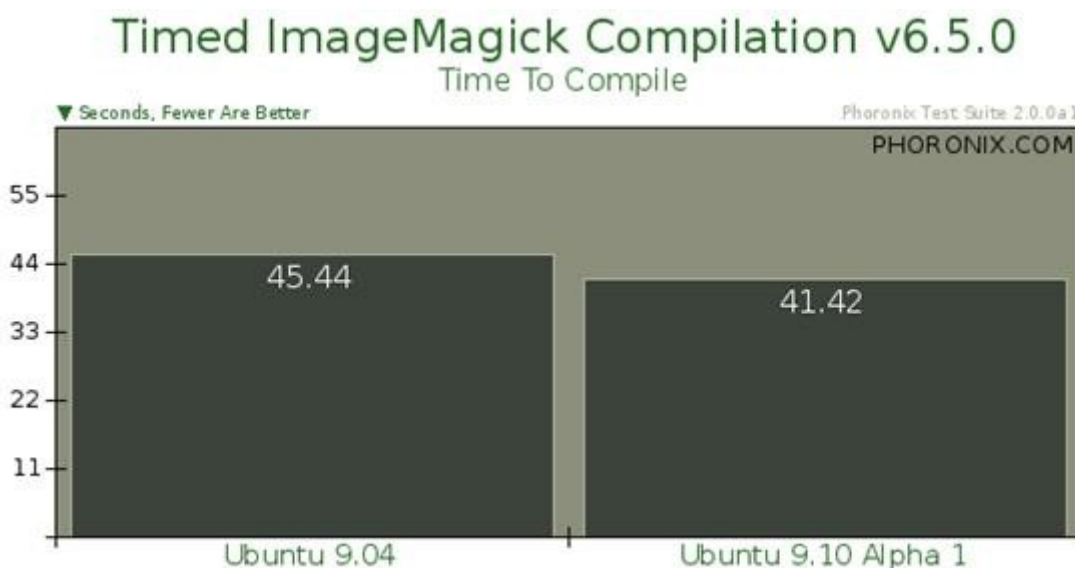
Ubuntu 9.10: 在性能飞跃中昂首起步

Karmic Koala Ubuntu 9.10 的第一个 Alpha 测试版已于昨日发布。虽然这只是最初的试验性版本, Plymouth 等激动人心的新特性也完全没有踪迹, 但我们依然看到了 Linux 2.6.30 kernel 内核、GNOME 2.27.1 桌面环境、GCC 4.4 编译器、UXA Intel 视频加速和新的电源管理技术等等, 而且它的性能表现绝对会让你大吃一惊。

此番测试的对象是 Ubuntu 9.04 x86_64 正式版和 Ubuntu 9.10 Alpha 1 测试版, 平台配置有 Core i7-920 OC 3.6GHz 处理器、华擎 X58 SuperComputer 主板、3GB CSXDDR3 内存、希捷 320GB 硬盘(ST3320620AS)、ATI Radeon HD 4770 显卡等, 测试软件依然采用 PhoronixTest Suite。



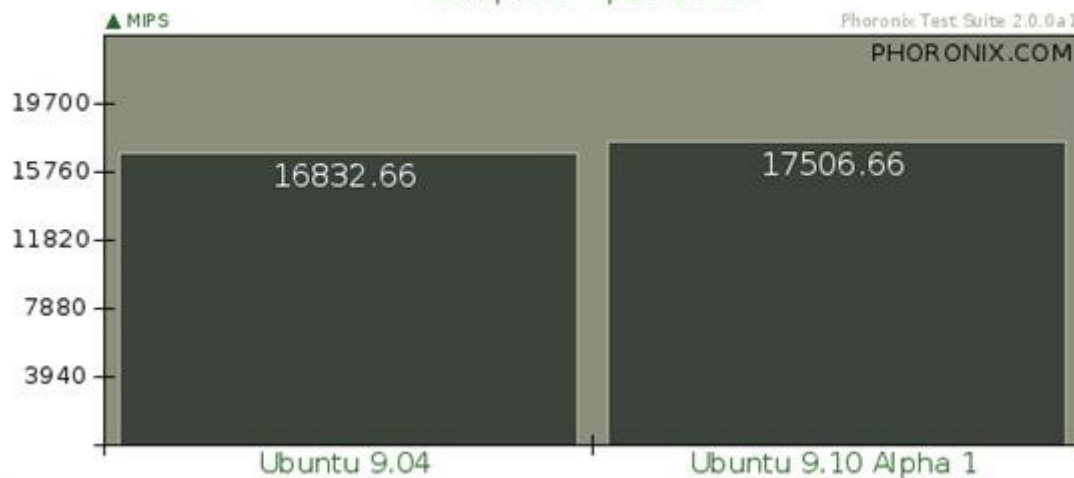
新版 GCC 4.4 编译器让 MySQL 编译时间缩短了将近 5 秒钟, 大约 6%。



ImageMagick 编译时间也缩短了 4 秒钟, 也就是 9%。

7-Zip Compression v4.65

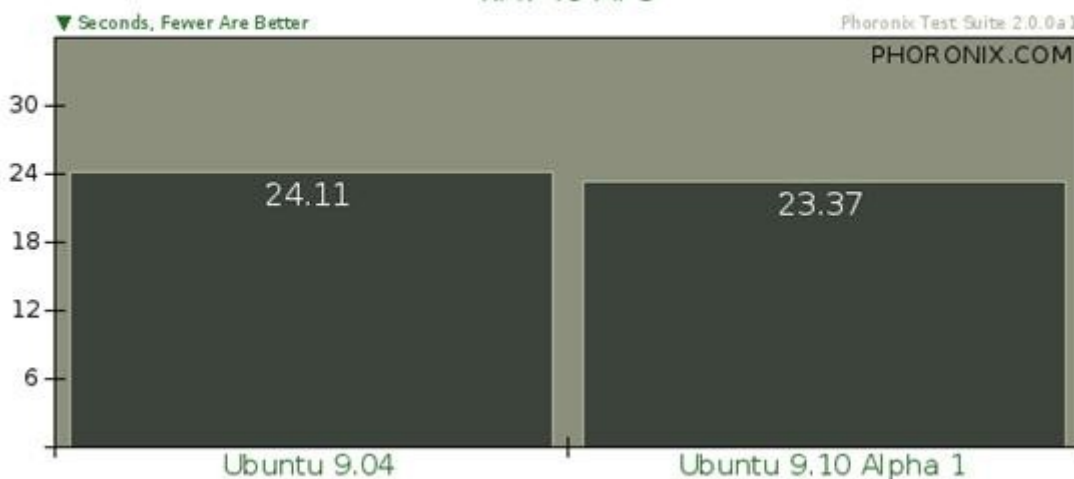
Compress Speed Test



7-zip 压缩速度加快了 4%。

LAME MP3 Encoding v3.98.2

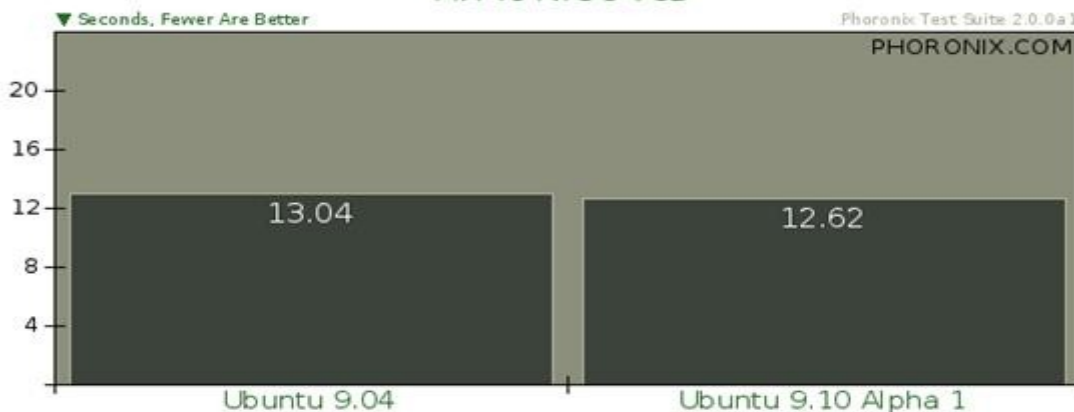
WAV To MP3



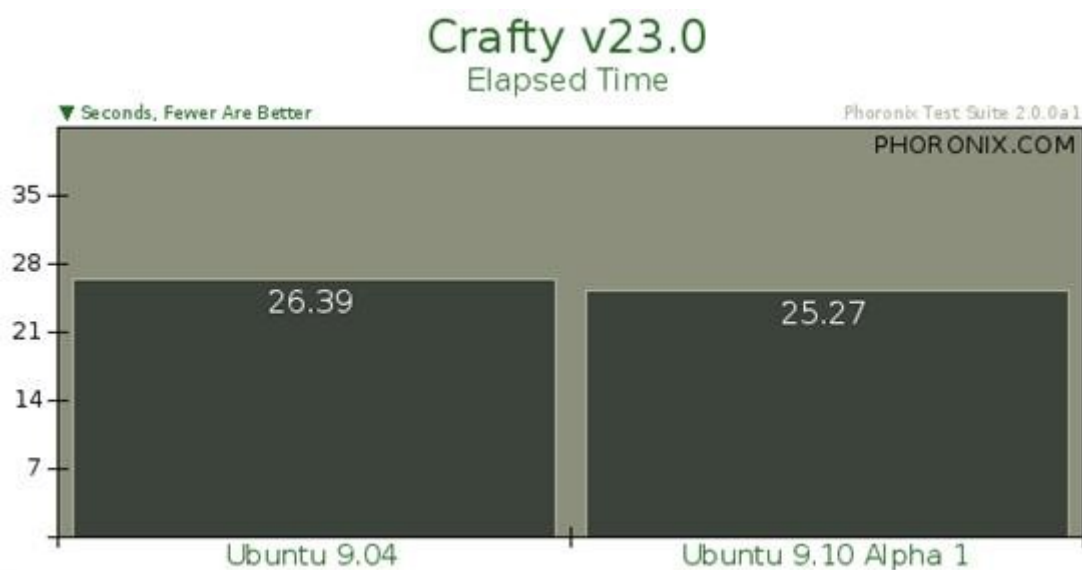
LAME MP3 编码速度加快 3%。

FFmpeg v0.5

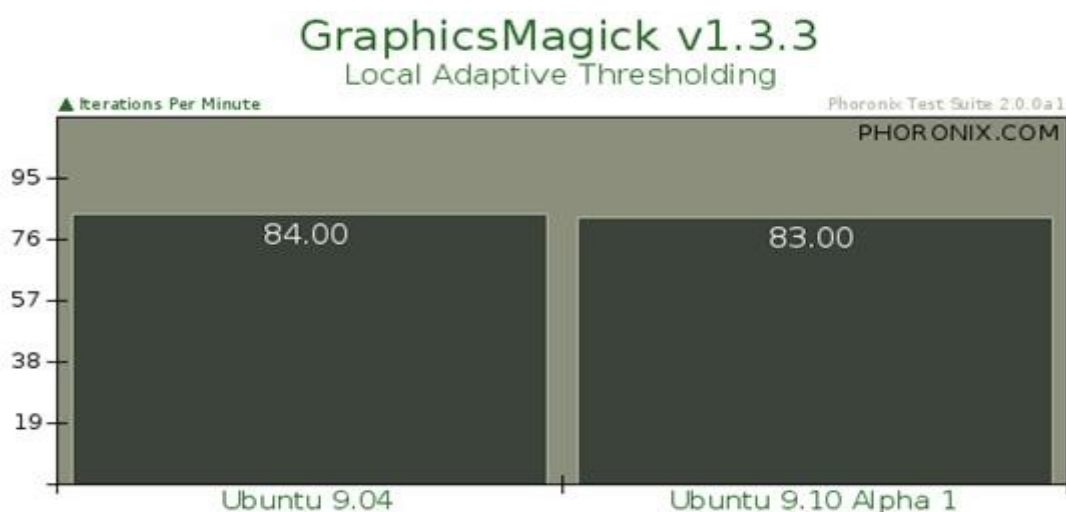
AVI To NTSC VCD



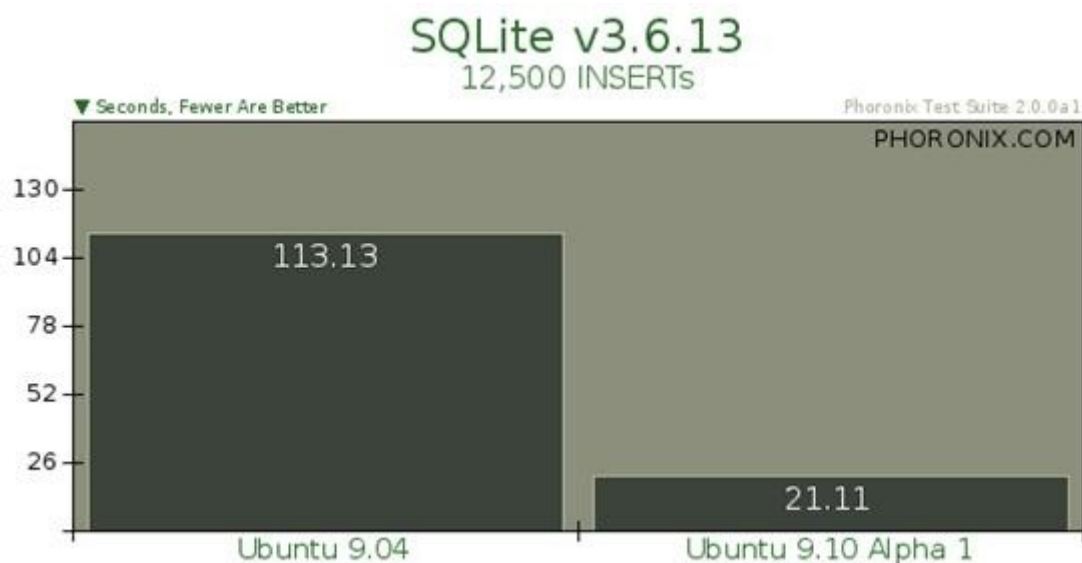
使用 FFmpeg 把 AVI 转码成 NTSC VCD 也有了 3% 的加速



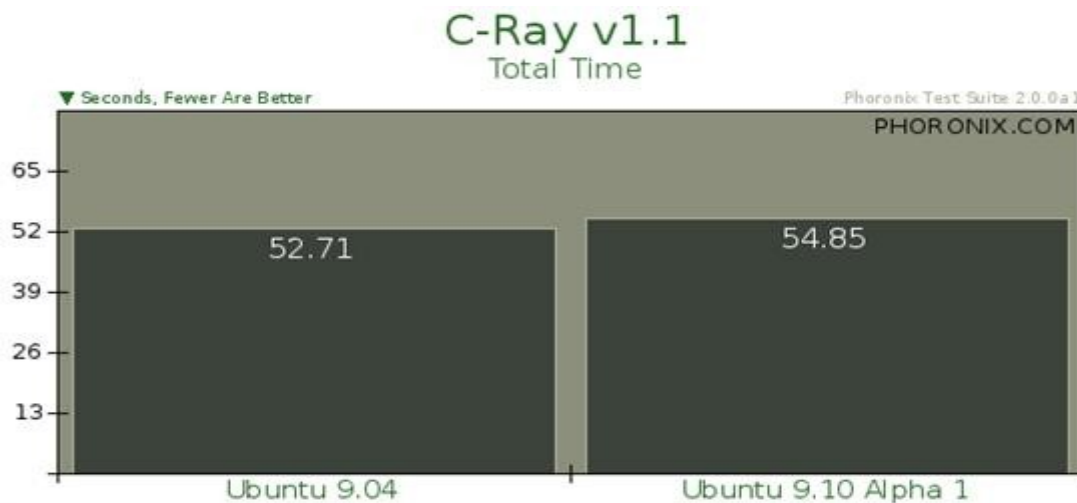
象棋引擎 Crafty 亦提速 4%以上



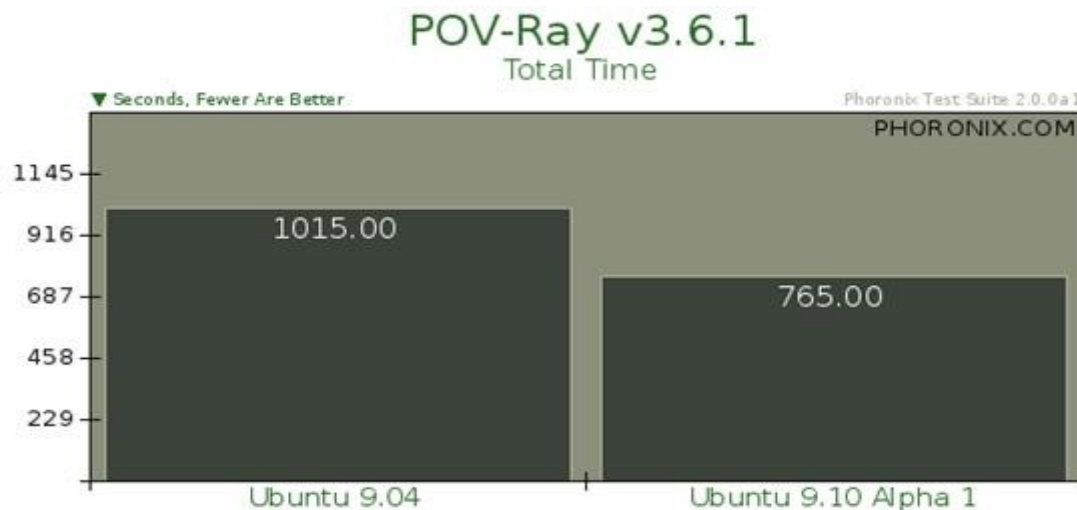
在基于 OpenMP 的 GraphicsMagick 里，Ubuntu 9.10 第一次落败，不过差距仅仅 1.2%



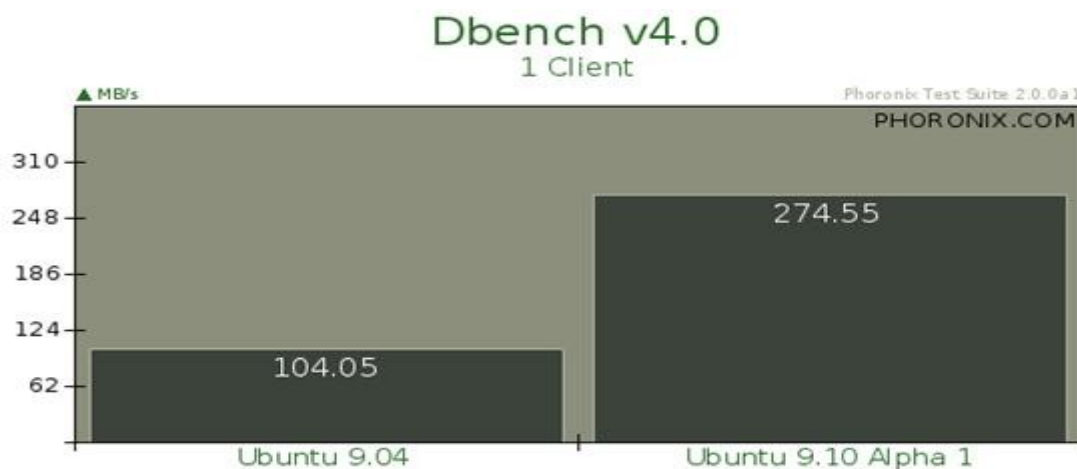
SQLite 性能在新系统里大幅提升，测试所需时间缩短了足有 92 秒钟，不到原来的五分之一，这显然要归功于 Linux 内核从 2.6.28 升级到 2.6.30



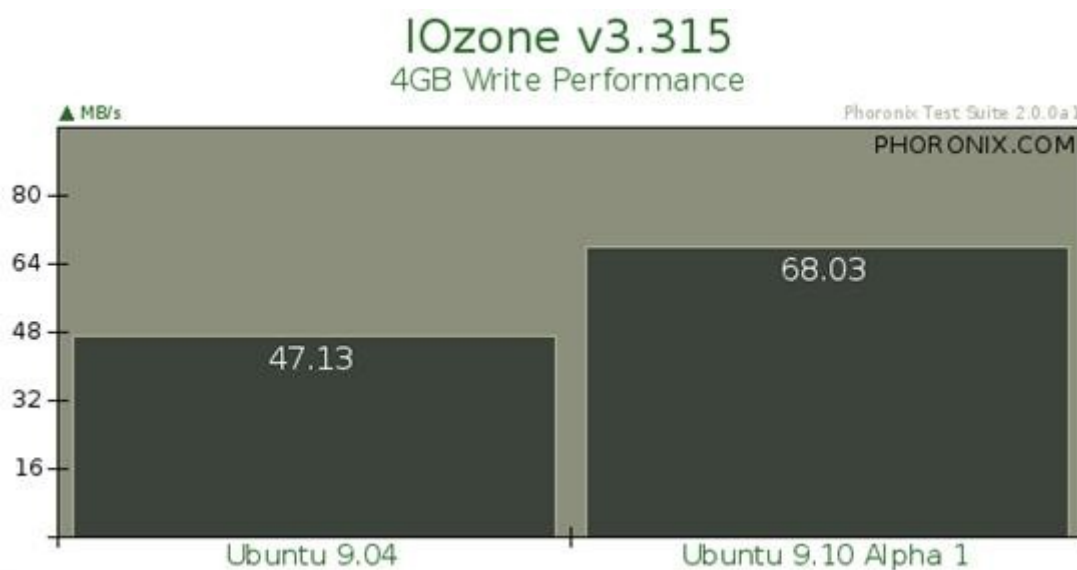
简单光线追踪引擎 C-Ray 在新系统里慢了 4%，超过 2 秒钟



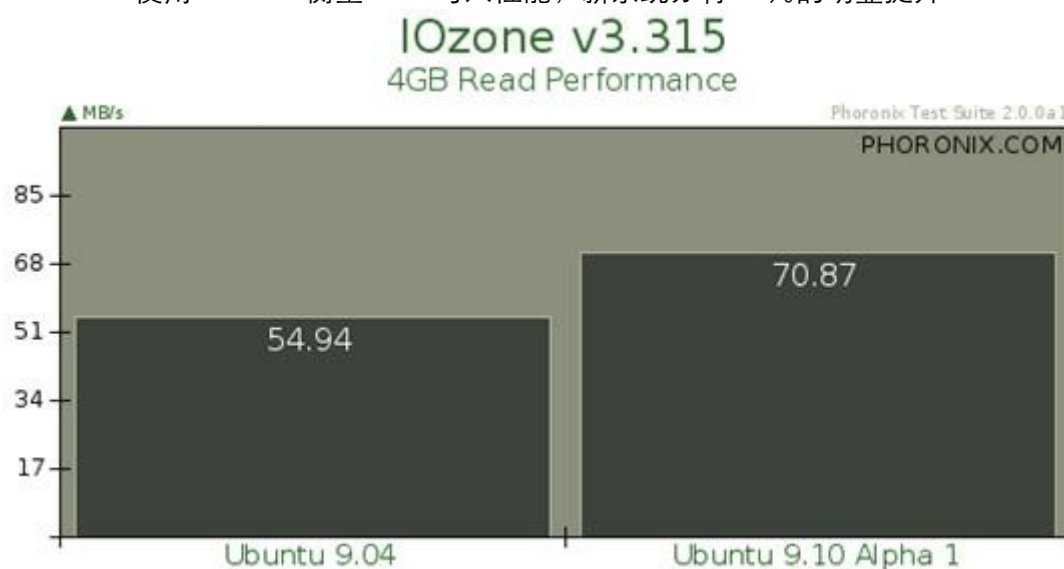
不过到了工业标准的 POV-Ray 里，新系统明显胜出，快了足足三分之一



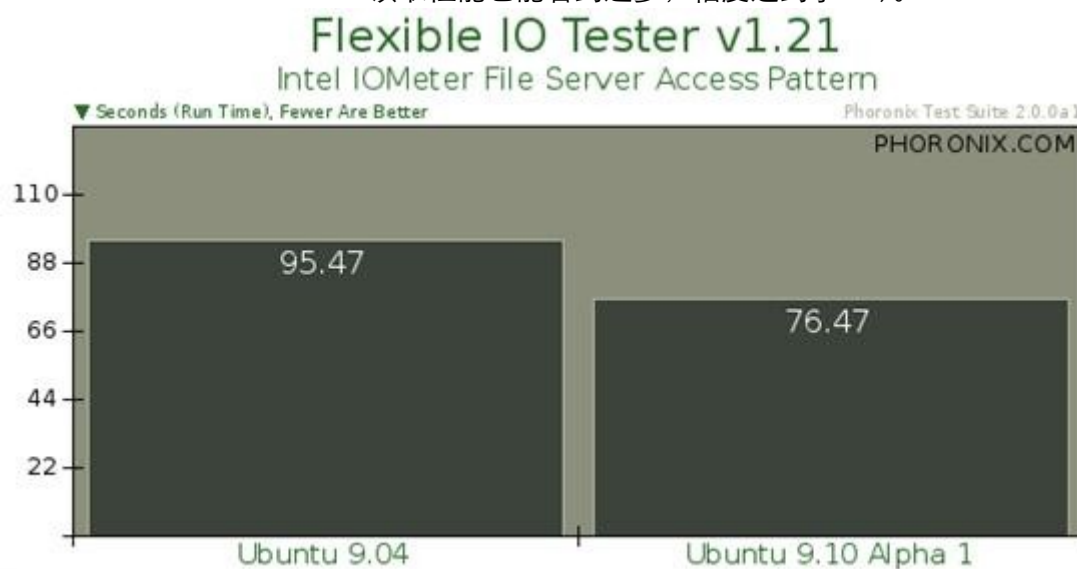
在均默认使用 EXT3 文件系统的情况下，新系统的 Dbench 磁盘性能达到了上个版本的 2.6 倍



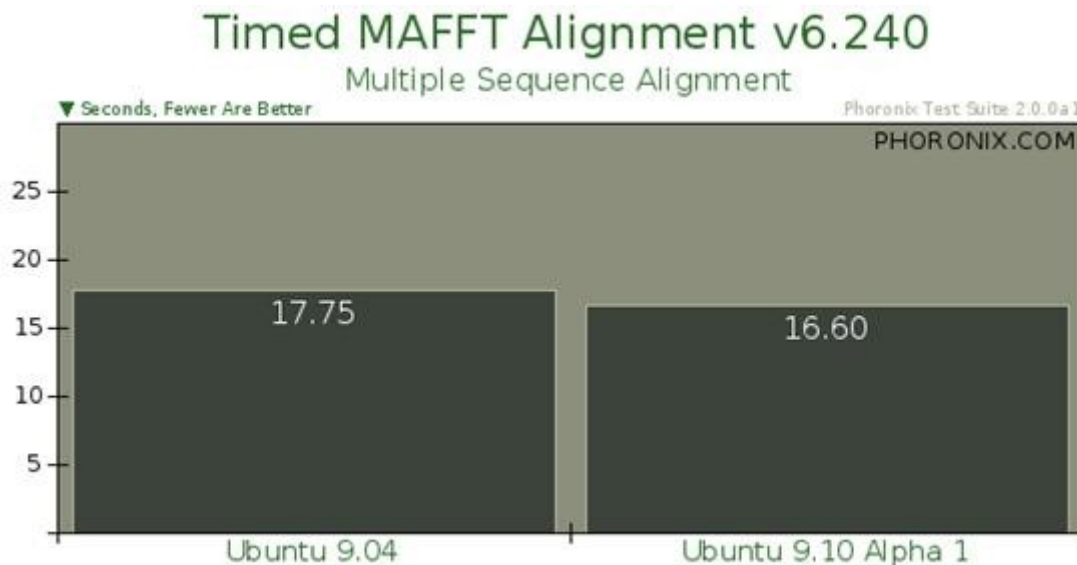
使用 IOzone 衡量 4GB 写入性能，新系统亦有 44% 的明显提升



IOzone 4GB 读取性能也能看到进步，幅度达到了 29%



另一项磁盘性能测试 Flexible IO Tester，新系统继续占优，耗时缩短整整 20%



最后一个项目 Timed MAFFT Alignment 里，新系统再次领先了一秒多钟

小结：

虽然按惯例来说 Alpha 1 版本不应该拿来来进行性能测试，但 Ubuntu 9.10 一登场就让我们惊喜异常。在测试的 15 个项目中，新系统取胜多达 13 个，带来了更短的编译时间(新编译器)、好得多的磁盘性能(新内核)和其它多方面进步。

当然也有迫切需要改进的地方，特别是 Intel Linux 图形性能，已经困扰用户很久了。现在说 Ubuntu 9.10 最终性能如何还为时过早，不过 Alpha 1 的表现让我们充满了期待，希望后边的版本能继续保持，同时加入、完善更多新功能，在终端客户体验方面也能有更好的进步。

MySQL 被分支已无法挽救？

在甲骨文宣布收购 Sun 一个月之后，MySQL 的未来仍然悬而未决。领先的轻量级开源数据库能否在领先的商业数据库供应商手中仍能保持欣欣向荣？目前看来形势不是很乐观。

在甲骨文收购之前，Sun 收购 MySQL 之后，MySQL 的主要开发者以各种理由纷纷离开 Sun，包括前 CEO Marten Mickos 和共同创始人 Monty Widenius。Widenius 还公开抨击 MySQL 的开发流程，称 Sun 的发布周期过于匆忙，质量控制不佳。另一位创始人 David Axmark 则因为厌恶公司的官僚作风和事无巨细的企业文化而辞职。在甲骨文收购 Sun 之后，MySQL 分支的开发已呈现公开化，主要为 Drizzle 和 MariaDB。上周 Monty Widenius 宣布成立开放数据库联盟（Open Database Alliance），但联盟成员没有甲骨文。现在的问题，正如前 MySQL 雇员现 Drizzle 开发者 Patrick Galbraith 所提出的，“MySQL 的哪一个分支将被认为是官方的？”，分支是开源社区的无法改变的现象，但通常只有一个能健康发展。甲骨文所希望只能是它没有站错地方。为了实现这一点，它将不得不努力争取开发者和社区的信任，放弃原来的一套，采取更开放的姿势。

行业观察

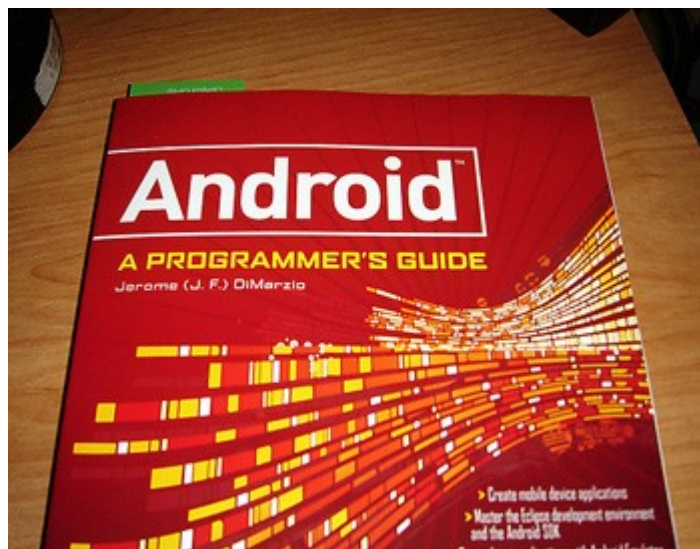
谁在打造 Android 的技术基石？

看好 Android 的人怎么说？

本月初，研究机构 StrategyAnalytics 发布报告指出，到 2012 年 Android 有望在全球智能手机出货量中占据 12% 市场份额，同时也会在其他设备中呈现增长。

资深分析师 BonnyJoy 得出的结论是，手机制造商巨头对于 Android 的支持是“十分有力”。他说，“显然，智能手机将会保持增长，而 Android 肯定会从这种增长中受益。”

上个月，设在伦敦的 Informa Telecoms&Media 发布了一个报告，预测到 2012 年 Android 智能手机的销售将会超过 iPhone。今年，全球手机总销售量将会降低 10.1%，但智能手机销量却将增长 35.3%。2008 年，智能手机销售量达到 1 亿 6200 万支，在历史上第一次超过了笔记本电脑的销量。



相对于蒸蒸日上的 Android，在 2008 年售出的智能手机中只有 49% 是基于 Nokia 支持的 Symbian 操作系统，明显低于 2007 年的 65%。Symbian 最近也走向开放源码，然而 Informa 的报告指出这项改变只能帮助 Nokia 将其领先地位保持到 2013 年。

同时，许多产业分析师都认为，Android 将会成为全球顶级的手机和 Netbook(上网笔记本)移动计算平台，已经对 Microsoft 构成了直接的威胁。

位于圣何塞的 Enderle Group 分析师 Rob Enderle 说，“Android 也会在 Netbook 上占有一席之地。把计算机的功能简化到只有上网功能和手机功能，这将是一股洪流，对于市场将是颠覆性的。许多设备制造商，包括 Hewlett-Packard，都在悄悄测试 Android。来自 OEM 的所有反馈都持“乐观其成”的态度。”

看起来，Android 将会在智能手机、上网笔记本和其他类似的设备中大行其道。

风河押宝 Android

在过去的 18 个月里，风河——这家总部位于美国 Alameda 的设备软件提供商指派了将近 200 名工程师，极力向全球移动计算设备市场推广 Android 操作系统。众所周知，Android 的背后就是 Google。

这项重大的行动显示，这家公司将大量资源集中于开放源码的消费电子移动软件市场是一项战略性的抉择。这是风河在航空航天与国防、电信网络和工业控制等行业取得成功之后的又一次飞跃。

风河在设备软件行业已经拥有 26 年的经验，从手机到电视机顶盒，从波音喷气飞机到火星探测车，他们的产品已经安装在超过 3 亿台设备之中。在过去的 20 多年中，风河长期保持着全球嵌入式操作系统及开发工具市场份额第一的地位，直到前些年被微软 Windows 取代而屈居第二。

2008 年 2 月，风河公司对公司管理运营架构进行了重组，划分为 VxWorks、Linux、开发工具和设备管理(DeviceManagement)四个部门。通过这次重组，风河将资源集中于发展最快速的市场——移动手持设备、车载电子设备、网络通信设备和航空与国防设备。

近年来，风河公司基于 Linux 的移动手持设备和车载信息娱乐设备领域增长十分迅速。特别是在手机 Linux 领域，不论是在以 Google 为首的 OHA 推出了 Android 平台上，还是在由众多手机制造商组成的 LiMo 基金会的 Linux 平台上，风河公司都扮演重要角色。

开源存储的现状

开源存储最近几年有很大的发展。备份、镜像、文件系统、NAS 和存储虚拟化方面出现了不少好的开放存储产品和解决方案，这使得用户不需要固态硬盘就可以将很多磁盘聚集起来以高性能水平运行。

EMC 存储平台高级总监 Jay Krone 表示：“EMC 通常是以 Linux 主机连接 EMC 存储产品的形式实现开源。用户购买基于 Intel 或者 AMD 处理器的服务器，然后通过配置 Linux 来最大程度上发挥硬件批量价格并将软件成本降至最低。” Krone 表示，用户倾向于向基于 Linux 的服务器中增加像 Apache Web 服务器这样的开源应用或者像 Oracle 数据库这样的专有产品来解决一系列业务问题。为了符合这种趋势，大多数 EMC 存储硬件和软件产品都进行了调整以更好地在 Linux 环境中运行。例如，EMC 的 PowerPath 产品线就是针对 Linux 环境的。

虽然 EMC 和其他一些数据存储厂商都意识到了这一趋势，但是他们对开源存储却抱有不同的看法。Digital 是一家运用大量 Linux 和 Sun 开源软件的公司，该公司首席技术官 Jason Williams 表示：“我仍然不会说市场中有很多开源存储应用。” Williams 表示，目前市场中领先的开源存储产品包括 Sun 的 ZFS 文件系统、用于备份的 Zmanda 和 Bacula、针对网络磁盘镜像的 DRBD。

StorageIO Group 创始人兼高级分析师 Greg Schulz 对开源存储的前景表现得更乐观一些。他说：“现在有很多不同来源的开源存储解决方案和应用，从卷管理工具、iSCSI 和 NAS 栈、文件系统、集群文件系统、基于目标的存储解决方案、重复数据删除和压缩等等，此外更不用说那些内嵌了开源技术的一体化解决方案和产品了。在传统的服务器和存储厂商中，Sun 可能是最突出，也是最极力宣扬开源存储的厂商。”

Sun 的 Amber Road 业务——包括 Unified Storage Systems (UFS) 或者 Sun Storage 7000 系列——就是基于预装了 OpenSolaris 和 ZFS 的 x86 硬件。这些设备支持文件和块层级的数据协议、自动精简配置、复制、镜像、快照、反病毒和分析工具。高性能版本中还增加了对 Linux 系统的支持。Sun 首席工程师 David Trachy 表示：“Amber Road 实际上是一款 NAS 系统，将廉价的服务器和开源软件集成是一款易于使用的应用。它的卖点就是可以帮你节省下购买专有磁盘系统的成本。

” Sun 的 Open Storage 产品线还包括 ZFS 文件系统、存储服务器和 JBOD 系统的 Storage J4000 产品线。Trachy 表示，这些产品方面的收入都有大幅的增长。尤其是 ZFS 表现格外抢眼。ZFS 提供了免费的 OpenSolaris、高层级的数据完成行以及站点之间的镜像功能。Trachy 表示，它可以被用于基础的大型数据存储库。这款系统还被 greenBytes 和 Nexenta Systems 等合作伙伴用于开发他们的存储系统产品。Trachy 表示：“不少初创厂商使用 ZFS，将其与 JBOD 集成以开发不同的产品和应用。现在 Sun 的开源产品线中欠缺的是光纤通道块层级存储和 pNFS，但是以后这些都将增加进来。”

除此之外 Trachy 还指出，ZFS 能够很好地兼容越来越受人们关注的固态硬盘。例如，Sun X4500 服务器用来自 STEC 的 ZeusIOPS 固态硬盘取代了 SATA 驱动器作为大容量内存卡。不过 SATA 仍然是大量数据存储的理想选择。在 ZFS 方面，Red Hat 的 Global File System (GFS)、Linux Logical Volume Manager (LVM)、ext4 和 BTRFS 文件服务器之间形成了激烈的竞争。美国明尼苏达大学率先开发出 GFS 用于针对 Linux 平台提供高性能和数据共享功能。虽然 GFS 是由 Red Hat 控制的，但是 LVM 已经在开源领域有了很多个版本。

开源存储项目

没有哪个大型数据存储厂商像 Sun 那样如此专注于开源技术，所以众多厂商分散于其他领域就不足为奇了。在备份领域，你会想到 Zmanda 和 Bacula Systems SA。

作为 Zmanda 的基础备份产品，Amanda 被认为是目前最受欢迎的开源备份和恢复软件，现在全球范围内 50 多万台运行不同版本 Linux、UNIX、BSD、Mac OS-X 和 Windows 的服务器和台式机都采用了这款软件。Zmanda 还推出了针对 MySQL 环境的 Zmanda Recovery Manager (ZRM) 软件。Zmanda 采用了与 Red Hat 相同的业务模式，但 Bacula 却是真正的前端开源技术。Bacula 管理来自或者向磁盘或者磁带的备份和恢复。Cleversafe 也是一家遵循开源业务模式的存储厂商。

开源存储发展的障碍

虽然市场中的开源应用不断如雨后春笋般不断涌现，但是开源存储的普及仍然面对很多障碍。Schulz 表示：“开源应该更多地被视为一个全面支持的解决方案——即使很多厂商已经提供了这样的解决方案——以此纠正人们认为开源只是针对那些成本有限的用户或者只是一个大型计算机科研项目的错误看法。”他还认为，应该以开源技术的最大特点——免费——将推动这项技术的更广泛应用。他说：“人们通常会认为，免费产品的价值低于付费产品，或者其价值和稳定性低于你可能购买的其他产品。因此，可能会存在对技术支持的担忧，或者误以为集成开源解决方案会增加成本和复杂性。”

Schulz 认为，阻碍开源技术普及的最大障碍可能是那些更基础性的问题。他认为，人们对开源技术在看法上的改变才能将这项技术提升到一个新的层级，而不是仅仅局限于过去那种“它不过就是开源”的简单的价值定位。Schulz 表示：“我希望了解的是，开源与其他解决方案在业务、成本、功能特性和价值定位方面的对比结果是怎样的。我真正希望听到的是，厂商在运用开源技术作为其整体解决方案和价值定位一部分的方面是如何做的。”

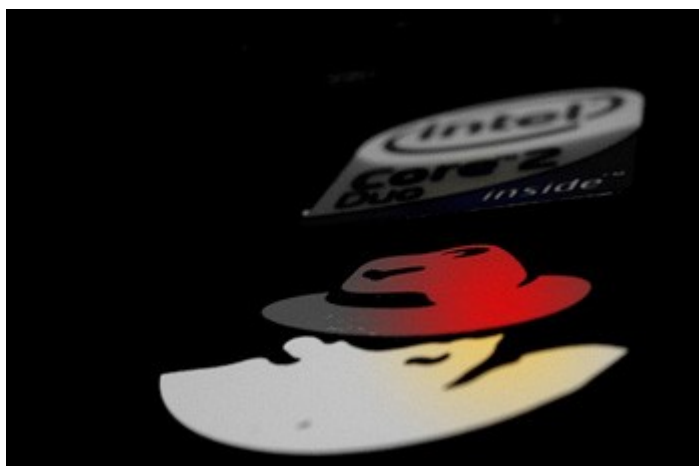
Comprehensive Consulting Solutions 公司的 Chip Nickolett 认为，我们应该看到开源技术最好的一方面。在他看来，说服核心存储专家的最大阻碍就是他们愿意在 SAN 或者其他昂贵的存储硬件上花费大笔资金，因为他们觉得从性能、数据完整性、备份和灾难恢复方面来讲存储是非常重要的，所以他们对采用开源存储节约几美元而可能面对潜在风险的做法并不感兴趣。他认为，在这种想法发生转变之前，开源技术只能挣扎在存储领域的边缘。Nickolett 表示：“我的确没有看到开源存储青睐人们多大的兴趣。虽然现在市场中有很多开源备份和磁盘管理工具、低端 NAS 和 SAN 产

品，但是这些在使用方面都没有成为‘关键’的一部分。”

到目前为止，Sun 和 Zmanda 都报告称其开源产品有很强劲的市场表现。Sun Open Storage 是 Sun 公司三月这个季度增长速度最快的业务。即使这样，这些产品在整个企业级数据存储市场中仍然只占到了很小的市场份额。

Red Hat 被收购不可避免 IBM 是最可能买家

近日，美国投资公司 Jeffries 分析师凯瑟琳·埃格伯特表示，尽管面临许多问题——来自甲骨文的竞争、收购传言和经济疲软，Linux 软件厂商 RedHat 似乎很“滋润”。但从长期来看，RedHat 将会被大公司收购——收购方最有可能是 IBM。



据国外媒体报道称，埃格伯特在一份投资报告中分析了 RedHat 的短期和长期前景。她指出，从短期来看，在经济疲软的背景下，RedHat EnterpriseLinux 和 Jboss 需求相当强劲，向政府机构的销售也将刺激 Red Hat 销售的增长。

市场研究公司 Piper Jaffray 分析师马克·墨菲也发表了观点相同的报告。墨菲通过对 70 家 RedHat 客户进行调查发现，RedHat 市场份额正在提高。

Red Hat 业绩非常好。华尔街预计，截至今年 5 月底的第一财季，RedHat 每股收益为 14 美分，营收为 1.717 亿美元。但 RedHat 业绩也存在数个变数，其中最大的是甲骨文收购 Sun，甲骨文会继续支持 RedHat 的产品吗？

尽管甲骨文首席执行官拉里·埃里森公开表示该公司将继续支持 Red Hat EnterpriseLinux，RedHat 内部弥漫着这样一种情绪：甲骨文更重视 Open Solaris 而不是 Red Hat EnterpriseLinux 是不可避免的，因为甲骨文会“保护”Solaris 维护业务营收。据估计，RedHat 三分之一的新业务来自 Unix 向 Linux 的迁移。RedHat 面临的危险是，甲骨文将向客户提供有吸引力的条件，让他们继续使用 Solaris。

埃格伯特表示，“考虑到 Red Hat EnterpriseLinux 对数据中心的重要性，我们相信 RedHat 被一家大公司收购是不可避免的——可能是 IBM。RedHat 成功的主要原因是，其软件相对便宜，Unix 软件能够方便地移植到 Linux 上等。RedHat 不是一家大公司，没有锁定客户的能力。大多数客户将 RedHat EnterpriseLinux 看作是打破大型厂商垄断的工具。甲骨文收购 Sun 后，RedHat 将面对两大巨无霸式的竞争对手，他们在定价方面都有着无穷的潜力，我们相信 RedHat 与甲骨文的竞争将越来越吃力。”

甲骨文能够提供廉价的软/硬件组合、很高的折扣，让客户继续使用或移植到 OpenSolaris，甚至向客户付费，不让他们使用 RHEL 以获得稳定的 Solaris 维护收入。RedHat 需要一家具有强大价格能力、良好的客户关系、有大量数据中心业务的合作伙伴，抵御甲骨文/微软的威胁，IBM 正合适。

IBM 能够帮助 Red Hat 抵御微软和甲骨文的威胁。IBM 收购 Red Hat 的重要意义在于：IBM 可以捆绑 RedHatEnterpriseLinux；Jboss 与 Websphere 相辅相成；IBM 可以得益于 RedHat 的开发社区；IBM 可以获得适用于各级基础架构的操作系统和虚拟工具；软/硬件捆绑越来越成为一种潮流，RedHat 将向 IBM 提供更多的软件工具。

MINIX 3 承诺比 Windows 或 Linux 更安全

阿姆斯特丹 Vrije 大学的计算机科学教授 Andrew S. Tanenbaum 正领导着一个开发 MINIX3 操作系统的项目，MINIX 3 的目标是比 Windows 或 Linux 更安全。

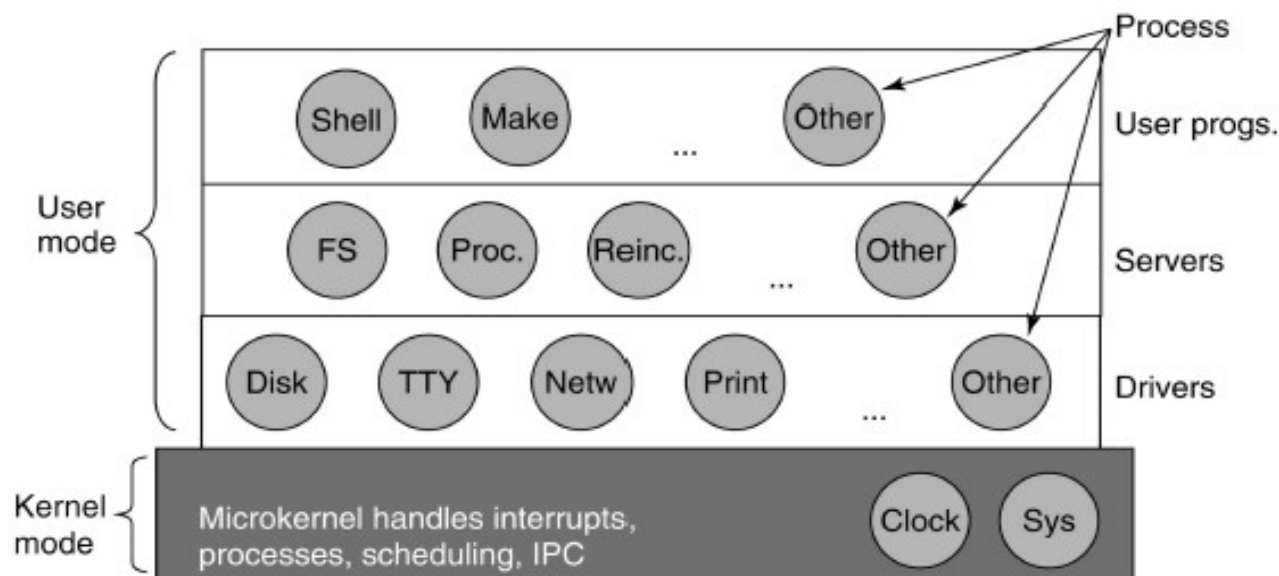
在他那份获得欧盟研究委员会（European Research Council）5 年 250 万欧元资助的研究计划书里，Tanenbaum 解释了为何他认为现有的操作系统不安全：

最严重的可靠性及安全问题是与操作系统相关的那些。核心问题在于现有操作系统都不符合 POLA——最低授权原则（Principle Of Least Authority）。POLA 说的是系统划分组件的方式，应当使必然存在于某个组件中的缺陷，不至于波及到其他组件。每个组件仅应该得到完成它本身工作所需的权限，不多不少。具体来说，它应该无权读写属于其他组件的数据，无权读取它自身地址空间之外的任何计算机内存，无权执行与它无关的敏感操作指令，无权访问不该访问的 I/O 设备，诸如此类。现有操作系统完全违反以上原则，结果就是造成众多可靠性及安全问题。

Tanenbaum 认为设备驱动程序是主要的肇事者：

典型的操作系统代码中大约有 70% 属于设备驱动代码。我们知道设备驱动部分的每行缺陷数量是其他部分的 3 到 7 倍。有据可查 63% 到 85% 的 Windows XP 崩溃根源是驱动错误，Linux 的情况也没什么理由会不一样。

MINIX 的操作系统架构稍有不同：



Tanenbaum 指出其中的主要区别是令设备驱动运行在用户态能提高安全性：

设备驱动处于内核之上，运行在用户态。每个驱动都作为单独的进程运行，受到内存管理硬件的严格约束，只能访问自己拥有的内存。设备驱动需要执行 I/O 操作的时候向微内核发出调用以获得服务（比如向物理设备发出指令）。微内核在执行任何调用之前，首先会检查该调用是否被许可。因此从声音驱动发出控制声卡的调用会被接受，而从声音驱动发出控制硬盘的调用会被拒绝。

总体目标是：

……将操作系统的结构重新组织为一个运行于内核态的微内核，外加若干用户进程去完成真正的操作系统任务。尽管一个微内核离完整的操作系统还很远，但这样的设计达到了我们的主要目标。我们的目标是将大部分操作系统代码移出到用户空间，这样我们就能将之划分成进程并阻止它们执行“危险”（控制）指令，内存管理硬件也能防止进程访问自身所属以外的内存。

目前 MINIX 3 的版本是 3.1.3a 版，它的主要特性包括：

- * 符合 POSIX
- * TCP/IP 网络
- * X Window 系统
- * 语言支持包括 cc、gcc、g++、perl、python 等等
- * 超过 650 个 UNIX 程序
- * 完全支持多用户和 multiprogramming 多任务
- * 设备驱动作为用户进程运行
- * 高度容错
- * 提供完整的 C 源代码。

由于 MINIX 实现了 POSIX 标准，大多数 UNIX 程序只要细微修改或无需修改就能在上面运行。该项目进入了 Google' s Summer of Code 2008，还设立了相应的 MINIX 讨论组。

开源技术：经济衰退期企业 IT 变革之道

信息技术是提供公司发展繁荣所需要的效率、生产力和竞争力的关键，因此，决策者应当注意 IT 领域中的变革。

研究公司 Gartner 预测，到 2012 年，90%的企业将使用直接的或以嵌入式的开源技术。开源技术并不是什么新东西，但它是企业决策者必须更加熟悉的术语。开源技术已经问世 20 多年了，为许多机构成功部署。在一些国家，它被应用在所有政府机构中，包括国防部门。

开源解决方案能够为企业难以忽视的机会，以降低 IT 的总成本，增加 IT 投资带来的回报。这已成为开源技术长久以来秘而不宣的事情。

开源解决方案是开发基于广泛接受的标准的高质量软件的创新的、社区使能的开发模型。根据通用公共许可证(GPL)授权的开源解决方案为客户提供了自由使用软件的灵活性，避免他们付出

被锁定在某一厂商的专有技术上的代价。Linux 操作系统是说明这种差别的开源技术的例子，而 Windows 则是专有技术的例子。

在当前的经济气候下，正当公司必须更多地利用技术来增加竞争优势之时，却到处都缺少投资新系统的资金。此外，作为减少成本的一部分，IT 人员被裁减，但与此同时企业对服务和支持的需要却并没有减少，甚至还在增加。低成本、高回报开源解决方案的价值主张变得比以往更具吸引力。

许多企业仍使用专有技术，有时是由于对开源的误解，但常常是由于厂商锁定的结果。这种情况正在发生迅速的变化。据在线新闻服务 CIO. Com 2008 年对 IT 和业务经理进行的调查显示，美国 53% 的企业使用开源技术，另有 10% 的企业计划在今后一年中部署开源技术。这种趋势在这部分世界同样明显。

没有行业能避免裁员、破产和市场需求减少的袭击。每家公司都必须认真研究 IT 开支，重新考虑它们的常规采购模式。今天的开源软件已经可靠、稳定和成熟得足以在最保守的主流企业中的关键任务系统中应用。

据 Forrester Research 说，企业目前将大约 80% 的 IT 预算用于保持运营和日常维护上。因此，投资创新、新硬件和软件的资金没有剩下多少。开源为企业提供了减少 IT 基础设施费用的机会。

促使大多数企业最初从专有软件迁移到开源软件的关键原因是省钱。在公司可以大大节省费用的同时，开源还提供给企业更多的东西。开源使企业可以按下启动许多涉及 IT 基础设施的活动的按钮。

※迁移到商用硬件。业界使用像 Intel x86 系统这类商用硬件而不是专有机器的能力代表着最大的节省费用的机会。不仅基础系统更廉价，而且维护硬件的成本也大大降低。

※消灭费用高昂的维护合同。专有软件的年维护费会大大增加这类系统的总拥有成本(TCO)。另一方面，开源产品的企业版一般作为核心价格模型的一部分包括更新和维护。

※提高工作人员的生产力。迅速的学习曲线以及可供使用的先进的管理工具使企业可以减少从事日常支持任务的人员的数量，把他们解放出来从事更战略性的工作。

※避免厂商锁定。企业常常发现自己受到软件厂商的许可协议以及维护合作涨价的束缚。开源软件采用标准技术开发，从而使开源产品的不同版本可以很容易互换。这就使企业避免了厂商锁定。Forrester Research 报告说厂商独立性对于 43% 的接受调查的企业用户“非常重要”。

※减少对专门的安全咨询师和工具的需要。由于开源系统，具体地说，Linux 操作系统比专有操作系统更安全，因此，保持系统和数据的安全需要更少的修补活动和资源。

仅凭这些理由，开源软件就是吸引力越来越大的选择。开源软件还走在时代的前面。协作开发是软件开发的开源模型的核心。大量的开发人员，常常包括来自世界各地的用户和厂商，合作开发保持开源软件处于技术前沿的功能性。开源产品常常是首先进入市场的具有创新特性和功能的产品。

现在是事关企业成败的关键时刻。在 IT 部门中，决策者面临困难的选择。他们可以采用减少成本的常规手段：裁员、控制资本开支和推迟新技术的部署。或者他们可以利用这一时刻真正改革他们的 IT 基础设施，放弃昂贵的专有 IT 解决方案，效仿 Amazon、NYSE-Euronext 和 Eveready 等机构采用低成本、高回报的开源解决方案。

专家专栏

参与社区开源项目 贵在坚持

-----Novell 中国区研发经理刘俊贤专访

ChinaUnix: 首先介绍下刘俊贤先生，刘先生英文名字叫 Alex Lau，网名 AvengerMoJo。从 96 年开始使用 Linux，刘先生现任北京 Novell Linux R&D 高级技术经理，从事 openSUSE 的开发与推广，曾任香港即时科技有限公司应用软件 R&D 技术总监，2003 年开始在国内多个重点城市建立及培训 Linux 有关课程，包括信息产业部的 1+1+1 项目课程研发和课本书写，与此同时他亦曾在北大、复旦和西交大学讲授 Linux 有关课程，除教学外他在北京建立了一个针对嵌入式 Linux 和多媒体应用的工作坊。Alex 毕业于美国北德州大学计算机工程系，曾担任美国北电网络 (Nortel Network) 从事软件分析和开发等工作，并热衷开放源码社区的活动。



ChinaUnix: 刘先生能给大家介绍一下 Novell 中国研发部门的情况么（主要工作、人员数量）？

刘俊贤: Novell 中国从 05 年把 openSUSE 引入中国后不停地拓展我们中国区的研发团队，从一开始三个开发人员到现在的五十多人。当中大部份都是在 Novell 的 Open Platform Solution(OPS)部门之内，故名思义，我们主要的工作是开发开源和 Linux 平台上的解决方案。

假如为 Novell 中国的开发项目做个分类的话，主要可以分为三个大类，一是服务器、二是桌面和三是 QA，当然这些部门中也有 SUSE Lab 的核心成员。另外，我们 OPS 内所有的研发人员都是直接对核心产品进行研究、开发、维护和测试的，而不会只做周边或低层次的工作。这也是我们研发团队和国内其他外企最大不同的地方，也可以说是我们的一个特点吧。服务器部分是我们标志性的服务器操作系统产品，在不同行业中都有大量的使用和良好的生态环境，而国内研发部门的主要工作是针对 HA 和虚拟化技术的研究和开发，其实 Linux 平台很适合作为 HA 的底层平台，因为它性能稳定，且可定制性强，可以针对不同的服务器应用而做出具有针对性的 HA 方案。虚拟化技术在市场上已经被炒到很火，我们的目标就给 SUSE 一个完美的 Guest 操作系统的称号，就是说你在任何平台上使用任何虚拟机，都可以很有效率的去使用 SUSE。

SUSE 桌面平台是市场上唯一的企业开源桌面解决方案，当中我们和其他跨平台的支持一定是 Linux 桌面上最好的产品，但是国内盗版导致的桌面生态环境严重被破坏，以致很难找到有第三方软件商和我们合作。但是近年来 Linux 桌面在全球的推广和其他新产品例如 Netbook 上都使用 Linux 去打开市场，其实我们从 05 年进入中国的时候就开始大力推动 SUSE 桌面和不同的第三方软件公司合作，包括微软。

对 Linux 预装的市场我们不但是硬件厂商商业市场上的首选，对 Netbook 的支持也是业内花最多人手和投入的，我们和 Moblin 的合作更进一步的说明了我们对于桌面市场上的信心。我们的桌面部门有一部分员工是在台湾的，主要工作就是桌面预装，产品测试和软件维护。预装方面主要包括三大难点，一是驱动的开发和整合，二是第三方软件的支持，第三点则是产品认证和售后服务。产品测试和认证都需要大量人手和时间，但桌面产品的周期很快很紧，自动化流程是我们 QA 部门的重要任务。当客户买了我们的服务后，和盗版不一样的是我们需要向客户提供软件的维护和安全更新。

而且很多人都不知道每当一个软件或内核需要更新时，我们都需要做大量的工作把有关的更新返回客户手上。所以 Novell 很认同开源的驱动并以减少维护成本，而且你可以看到有不少开源驱动都是由 Novell 帮助硬件厂商推动而成的，这一方向我认为我们的研发人员对开源驱动的贡献实在是很大的。

ChinaUnix: 能介绍一下 Novell SuSE Linux 目前在中国的发展情况，以及 openSUSE 在国内的推广情况。

刘俊贤: 现在国内以 IDC 的报告来看，SUSE Linux 在企业服务器有领先的优势，而 Linux 桌面操作系统的全部表现才占到 1 个百分比的桌面操作系统市场，所以说明不了什么问题。但从 Novell 眼中有两方面希望外界了解的：一，我们会保持我们在大客户上的专业优势，Novell 是微软的合作伙伴，所以各方面的企业 SI 和 ISV 可以很容易的成为我们的合作对象；二，我们希望扩展渠道销售的能力，以加强我们在中小企上的影响力。当然我们已经努力不懈的把很多预装 Linux 的机会建立起来，以便将来对用户进行软件硬件认证的支持和帮助提升客户服务质量。

Novell 在国内目前还没有一个专门针对 openSUSE 发展的部门，openSUSE 的本身是社区，我们会进一步开放我们的系统给开源贡献者使用，而有志参兴的朋友可以加入我们 openSUSE 的官方网站。我们在北京每月也有一个针对开发者的活动，可以到开源聚贤(<http://bit.ly/OXwlb>)参与。

ChinaUnix: 最近 openSUSE Build Service 加入 Linux 开发者网络，能给我们谈谈将来的发展，还有它对 Linux 世界的影响。

刘俊贤: 打开 SUSE 的社区，本来就是要从 openSUSE 做起，BuildService 就是一个很好的切入点，社区的建立最基本的重点是给大众发挥自己能力和加入优秀组织的机会。BuildService 不但可以给你机会为你喜欢的开源软件打包和维护，你同时可以为不同的发行版本建立安装包和构建你的开发和测试团队，最终你可以建立自己的发行版本。

ChinaUnix: 对接下来 5 月份即将开始的 openSUSE 社区周来说，openSUSE 团队有何打算，在中国又想怎么展开和推广？

刘俊贤: 我在上面也说了一些，openSUSE 希望可以更大力地去开放我们的资源，给社区更大的权力和义务，把 openSUSE 建立成为更独立和不依赖 Novell 的开源社区。在国内的整体环境还有很大的改进空间，而参与者不足是一个致命伤，对我而言重点还是希望更多的人了解 openSUSE 上所提供的服务，例如 BuildService 等工具，所以我们也计划在国内推出开发者发行版本，大家可以关注开源聚贤网站。

ChinaUnix: 能给国内开源技术人员和学生提出一些建议，如何参与开源？

刘俊贤: 参兴开源和做社工一样，没有什么特别的需要，只要你有心想就可以了，只要你找到感兴趣的方向发展，还有是恒心和方便自己的生活习惯（下面我会再附加说明），就可以在开源社区中成长。开源的工作从我身上可以分为两类，第一种是向社区的贡献，二是工作的需要。二者是不一样的但不冲突。当我面向社区时，我不会计较收入多少，而更看重的是对社区的发展，我自身能力的加强，人脉的扩展，社交能力的提升，网络不同文化交流的经验和推广并维护团队的精神。第二种是我的专业工作，和其他在非开源公司工作的朋友有所不同的地方是，我可以看作对开源作贡献，但不一样的地方是，我的本质是希望公司从中获益。我也知道在国内很多公司都打着开源的名义但做私有软件的工作，只要没有做非法的事（例如不守许可证、盗版等行为），我并不会认为这些公司做错，只是不道德。同时也有一些公司是说得很明白，员工不可以参与开源活动，所以大家如果有心一边工作（开源与否）一边参与开源，就必需先作选择，不然到时候你想参与也没有办法。同时

如果你的专业工作并非开源，千万不要参与同类开源项目，以免瓜地李下到时候被其他公司告就不好了。最后就是你生活习惯方面，很多人少看了这一点，就如做运动一样，有专业运动员的时间表，也有非专业的运动时间表，首先你要对你的兴趣做评估，一天能花多少时间？一周、一个月或一年能做多少东西，然后再把你的目标定立成为时间表，并对自己进行奖励和惩罚，当然你可以找同伴或多告诉你身边的人以作支持，如果你没有办法持之以恒，就如很多人做运动一样，二三周后你就会发现你没有动力再做，然后不了了之。国内外很多人把开源重点都放在技术的层面上，由如以往人材看 IQ 一般，但做开源没有所谓高手低手，只有成功和失败的项目，而成功与否很多时间不是代码技术上的区别，而是推动项目的社区吸引人材的能力问题，所以我对国内有心想参与开源的朋友最要搞清楚自己的"选择"。

ChinaUnix: 作为一个普通的 Linux 和开源技术人员，我们如何推广 Linux 和开源，让更多的人来了解它？

刘俊贤: 你是问我是吗？：）我作为一个普通开发人员，我已经做了很多有关开发方面的推广，大家可以用作参考，例如开源聚贤工作坊。作为一个用户我也有参与北京 Linux 用户协会 <http://www.beijinglug.org/>。我现在也构想以一个普通用户的身份作主题去写一本小说，从中说出 Linux 和开源对日常人有什么影响和经历。而日本已经有以 Linux 为主题的漫画。所以我认为可以做的事太多，你的选择你作主吧。

ChinaUnix: 今天 google 的 SOC，Novell 打算怎么参加呢，想要解决哪些问题？

刘俊贤: 我们每年都有参加，上一年我是其中一个项目的 Mentor，就是人脸识别的登录，<http://code.google.com/p/pam-face-authentication/>。很不错的项目，一般来说我们会在 idea.openSUSE.org 上的东西和不同 Mentor 的想法在网上发报，学生可以针对个人的喜好参与，同时你也可以向 openSUSE 提交自己的想法。今年我们有针对 ARM 平台上的移植，MIPS 平台（龙芯）上的移植，对 A11Y 上的支持，对 openSUSE 网站上使用 openID 和加强 BuildService 的 git 支持等等。大部分的项目都是和 openSUSE 有直接和间接的关系，同时可以让更多人使用到 openSUSE 的平台。

ChinaUnix: 现阶段虚拟化技术如此大热，能简单地谈一下 Linux 平台下的虚拟化技术的发展及趋势么？

刘俊贤: 上面我简单提到，我不是这方面专家，但 KVM 和 Xen 等不同技术方向都有它们的优劣，所以我相信成为最好的"Guest"是一个很好的目标。

ChinaUnix: Novell SuSE Linux Enterprise 11 一个月前发布，有什么新的功能和特性？能给大家简单的说一下么？

刘俊贤: 桌面方面我们加强已有的三维效果，对多媒体的支持上加上 Moonlight（Silverlight 的开源版）和能直接播放 WMA 的编码（从微软的网站自动下载解码器）。当然不能少的就是新版的 Gnome2.24 和 KDE4 桌面，我们的大牛 David Reveman 针对 rdesktop 加上三维功能，可以给 remote desktop 加上三维效果同时把 Thin Client 和虚拟化桌面上的解决方案加上了一大步。OpenOffice.org 的 3.0 版本有更强的兼容性，在 VBA 的支持上更加符合企业的需要。

ChinaUnix: 如何看待时下热门的上网本 Linux 市场，以及 Novell 的 SuSE Linux 在这方面将有何打算？

刘俊贤: 我们非常看好网本的 Linux 市场，有关这方面的策略不便现在透露，大家可以静观下半年的市场变化。当然我们和 Intel 上 Moblin 的合作已经提到台面，我们 OPS 对上网本的支持并不少，主

要是加强上网功能，加快始动速度和应用的友好性。为了迎合市场的要求，我们可能在下半年开始加快我们以往二年发行新版本的周期，并以 openSUSE 的基础提供半年一新版本的桌面发行周期，具体细节还没有完全定下来，主要是为质量、更新和安全升级维护的因素做更多的考虑。

ChinaUnix: 好吧，不能免俗，能谈谈最近的 Oracle 和 Sun 的并购案么？对开源世界会有什么影响？

刘俊贤: 我很难以公司立场去说个人观点，同时我是搞技术的不是搞占卜的，很难给你一个很好的说法。但我可以说出一些现实的事情，就是 Sun 公司有很多开源的项目，而对 Oracle 而言不一定有它的迫切性和投资回报，所以从客观的层面来看，这些项目是会有一定的影响。但在这些事都没出现以前我还是不能莽下定论。

技术新知

基于 Nessus 的主机网络安全检测

ChinaUnix 网友: kns1024wh

Linux 主机网络安全检测

采用基于 OpenSource 的 Nessus 实现对 Linux 主机网络安全检测, Nessus 是一个 OPENSOURCE 的功能强大而又易于使用的远程安全扫描器。安全扫描器的功能是对指定网络进行安全检查, 找出该网络是否存在有导致对手攻击的安全漏洞。Nessus 系统架构被设计为 client/sever 模式, 服务器端负责进行安全检查, 客户端用来配置管理服务器端。检查的结果可以 HTML、纯文本、LaTeX (一种文本文件格式) 等几种格式保存。

在 Nessus 客户端, 用户可以指定运行 Nessus 服务的机器、使用的端口扫描器及测试的内容及测试的 IP 地址范围。Nessus 本身是工作在多线程基础上的, 所以用户还可以设置系统同时工作的线程数。这样用户在远端就可以设置 Nessus 的工作配置了。安全检测完成后, 服务端将检测结果返回到客户端, 客户端生成直观的报告。在这个过程当中, 由于服务器向客户端传送的内容是系统的安全弱点, 为了防止通信内容受到监听, 其传输过程还可以选择加密。

Nessus 由客户端和服务端两部分组成。使用 Nessus 首先需要完成服务器端的安装及设置。

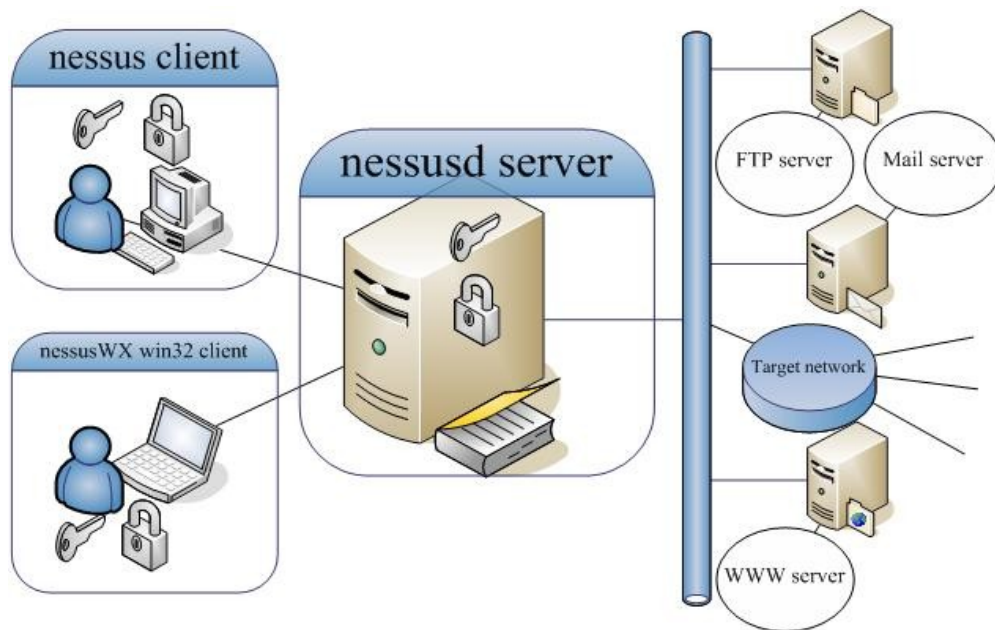


图: Nessus 应用架构图

安装 Nessus 安装步骤

Nessus 服务器端 RPM 包安装

```
[root@dr ~]# rpm -ivh Nessus-3.2.1-es5.x86_64.rpm
Preparing...
##### [100%]
 1:Nessus
##### [100%]
nessusd (Nessus) 3.2.1. for Linux
(C) 1998 - 2008 Tenable Network Security, Inc.

Processing the Nessus plugins...
[#####]

All plugins loaded
- Please run /opt/nessus/sbin/nessus-adduser to add an admin user
- Register your Nessus scanner at http://www.nessus.org/register/ to obtain
  all the newest plugins
- You can start nessusd by typing /sbin/service nessusd start
```

Nessus 安装到/opt 目录中，目录信息如下：

```
[root@rs3 nessus]# pwd
/opt/nessus
[root@rs3 nessus]# ls
bin com etc lib man sbin share var
```

创建 Nessus 用户

Nessus 服务端有自己的用户资料库，其中对每个用户都做了约束。用户可以在整个网络范围内通过 nessusd 服务端进行安全扫描。Nessus-adduser 是 Nessusd 的附带工具，安装完毕后，在安装目录下会产生这个程序。

创建用户的方法如下：

```
[root@rs3 snort]# /opt/nessus/sbin/nessus-adduser
Using /var/tmp as a temporary file holder
Add a new nessusd user
-----
Login : admin //输入用户名
```

```
Authentication (pass/cert) [pass] : pass //选择认证方式
Login password : //输入口令
Login password (again) : //输入口令

User rules
-----
nessusd has a rules system which allows you to restrict the hosts
that admin has the right to test. For instance, you may want
him to be able to scan his own host only.

Please see the nessus-adduser(8) man page for the rules syntax

Enter the rules for this user, and hit ctrl-D once you are done :
(the user can have an empty rules set)
Login      : admin
Password    : *****
DN          :
Rules       :
Is that ok ? (y/n) [y]
user added.
```

配置 Nessus 服务端程序 Nessusd

它的配置文件为 `nessusd.conf`，位于 `/opt/nessus/etc/nessus` 目录下。一般情况下，不建议你改动其中的内容，除非你确实有需要。

启动 nessusd

在上面的准备工作完成后，以 root 用户的身份用下面的命令启动服务端

```
[root@rs3 nessus]# service nessusd start
Starting Nessus services:          [ OK ]
```

确认 Nessus 已经启动，查看日志信息

```
[root@rs3 nessus]# tail -f /opt/nessus/var/nessus/logs/nessusd.messages
```



```
[Mon Aug 25 10:25:22 2008][11024] nessusd 3.2.1 (build A919) started
```

确认 Nessus 已经启动，检测进程状态信息

```
[root@rs3 nessus]# ps -ed | grep nessus
11024 ?    00:00:00 nessusd
```

确认 Nessus 已经启动，检测网络端口状态信息

```
[root@rs3 nessus]# netstat -ant | grep 1241
tcp  0  0.0.0.0:1241 0.0.0.0:*  LISTEN
tcp  0  0:::1241 :::*  LISTEN
```

安装 Nessus 客户端

为便于扫描操作，可以将 Nessus 客户端安装在一台 Windows XP 上，并开放 Nessus 服务端主机的 1241 端口。

开放 1241 Nessus 服务器端口

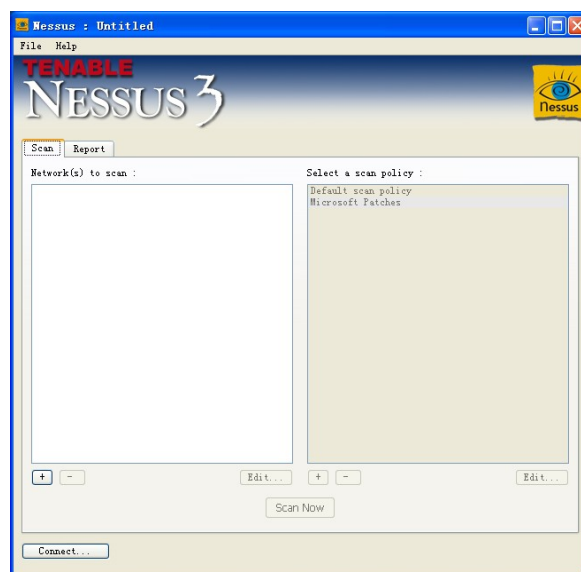
```
[root@rs3 nessus]# iptables -I INPUT -p tcp --dport 1241 -j ACCEPT
```

安装 Nessus Windows Client（具体过程略）。

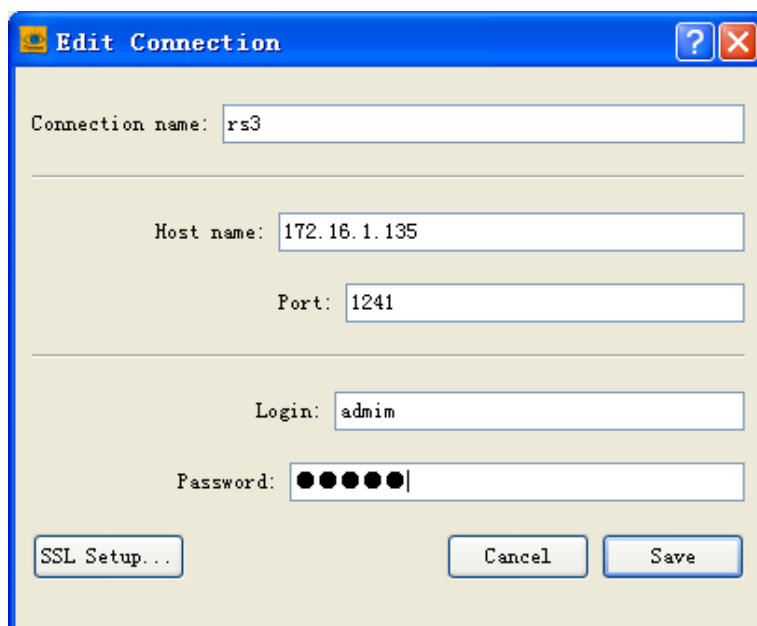
执行 Nessus 进行安全扫描

按照上面的方法启动 Nessus 的服务进程后，就可以运行 Nessus Windows Client 客户端程序开始安全扫描。

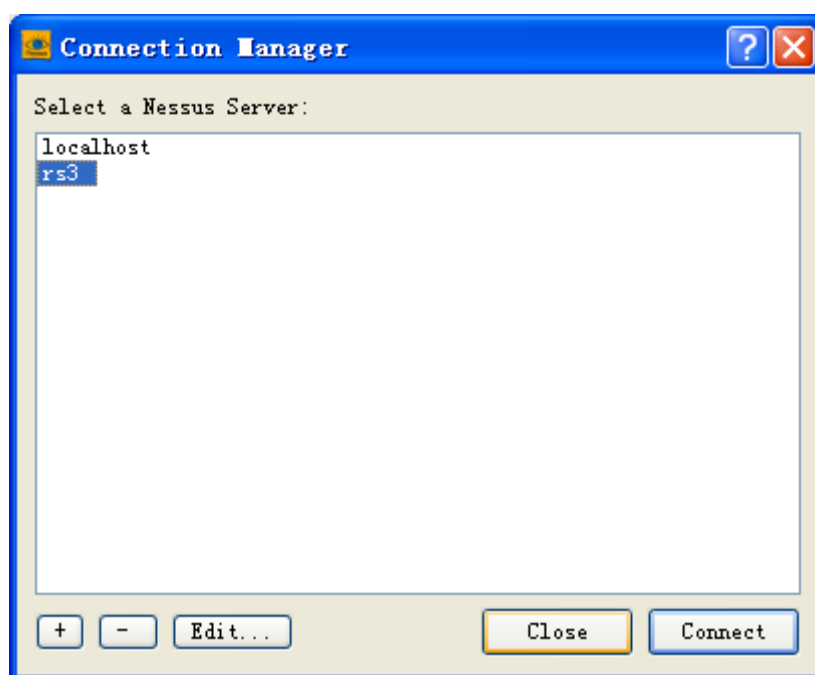
运行 Nessus 客户端,建立与服务器的链接



图：选择 Connect 链接到您选择的服务器

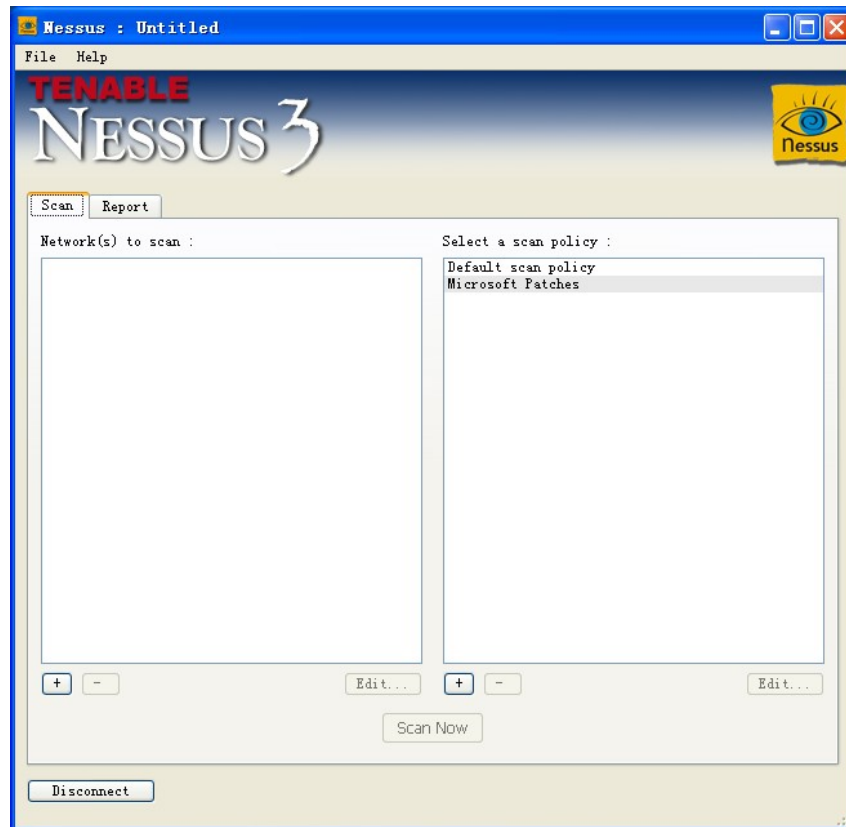


图：配置 Nessus 服务端连接信息



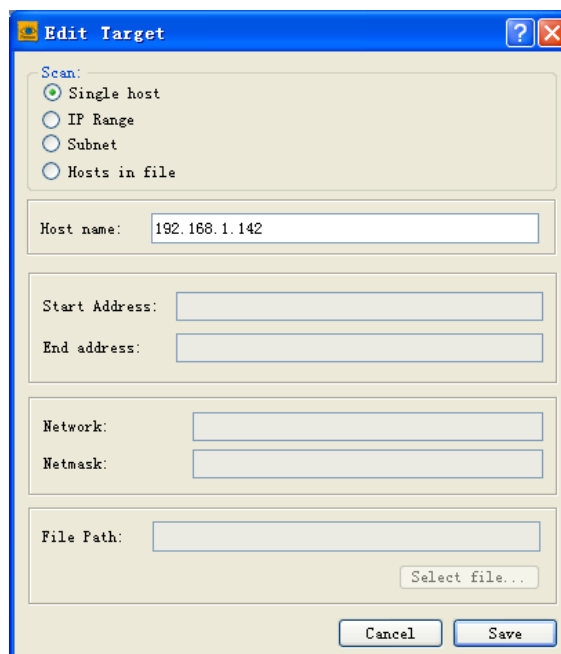
图：设置好的 Nessus 服务端连接信息

选择 connect 进行连接



图：已经与 Nessus 服务端建立连接

添加需要检查的目标服务器

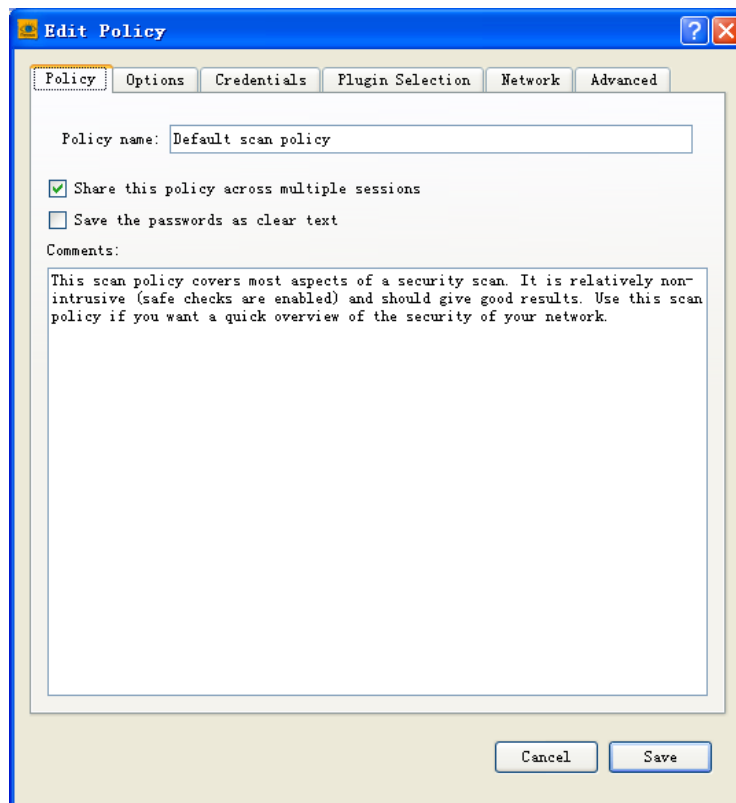


图：输入要检查的目标机器地址

在这个里面可以根据选项自行选择是扫描多台目标机器或者单台.然后选择 save .

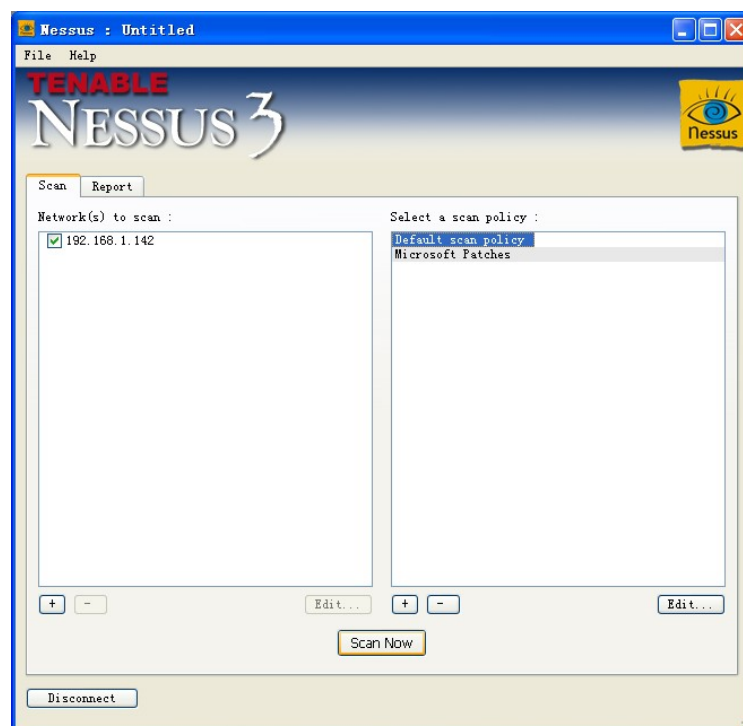
配置本地扫描规则

在 Nessus 客户端界面中点击右面的 select a scan policy 栏目中的+ 添加规则.



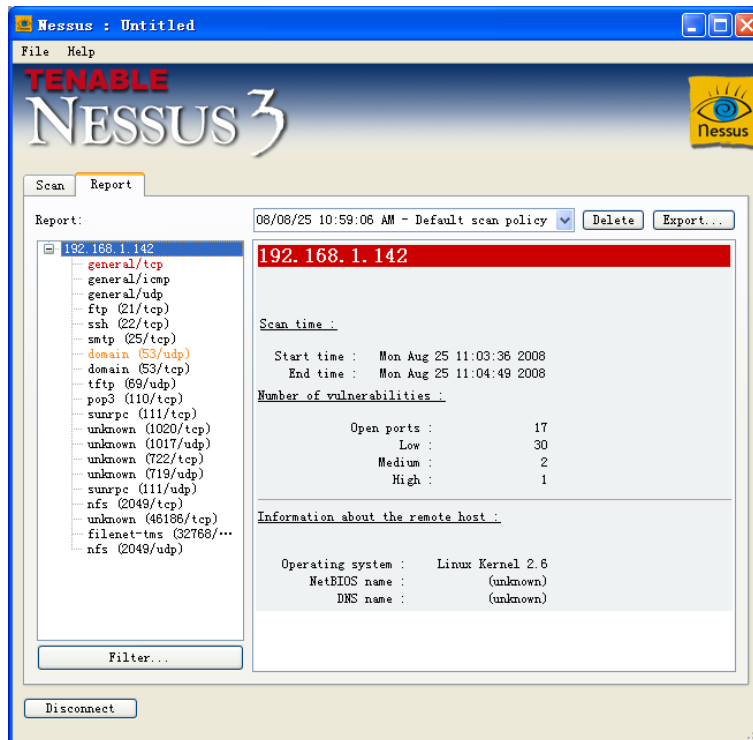
图：编辑策略 Policy name 可根据自己需要自行添加.

进行扫描



图：进行扫描

选择刚刚建立好的扫描策略以及目标服务器的地址.点击 Scan Now 进行扫描.



图：扫描报告

通过 Nessus 运行日志，确认 Nessus 运行状态



```
[root@rs3 nessus]# tail -f /opt/nessus/var/nessus/logs/nessusd.messages
[Mon Aug 25 11:35:59 2008][4496] gallery_script_exec.nasl (process 2041)
finished its job in 0.000 seconds
[Mon Aug 25 11:35:59 2008][4496] user admin : launching
aardvark_422_remote_file_include.nasl against 192.168.1.206 [2043]
[Mon Aug 25 11:35:59 2008][4496] aardvark_422_remote_file_include.nasl
(process 2043) finished its job in 0.000 seconds
[Mon Aug 25 11:35:59 2008][4496] user admin : launching
nukedit_email_sql_injection.nasl against 192.168.1.206 [2044]
[Mon Aug 25 11:35:59 2008][4496]
phpdocumentor_1_3_remote_file_inclusion.nasl (process 2042) finished its job
in 0.000 seconds
[Mon Aug 25 11:35:59 2008][4496] user admin : launching
movabletype_320.nasl against 192.168.1.206 [2045]
```

导出扫描报告，并对 nessus 扫描报告进行分析，查看 Nessus 导出报告：



List of hosts	
192.168.1.1	High Severity problem(s) found
192.168.1.2	High Severity problem(s) found
192.168.1.3	High Severity problem(s) found
192.168.1.4	High Severity problem(s) found
192.168.1.7	High Severity problem(s) found
192.168.1.8	High Severity problem(s) found
192.168.1.9	High Severity problem(s) found
192.168.1.10	High Severity problem(s) found
192.168.1.11	High Severity problem(s) found
192.168.1.12	High Severity problem(s) found
192.168.1.13	High Severity problem(s) found
192.168.1.14	High Severity problem(s) found
192.168.1.15	High Severity problem(s) found
192.168.1.17	High Severity problem(s) found
192.168.1.18	High Severity problem(s) found

图：报告中出现的 High Severity problem(s) found 需要详细检测此主机的报告信息



List of hosts	
192.168.1.142	High Severity problem(s) found

[^] Back

192.168.1.142

Scan time :
Start time : Mon Aug 25 11:03:36 2008
End time : Mon Aug 25 11:04:49 2008
Number of vulnerabilities :
Open ports : 17
Low : 30
Medium : 2
High : 1

Information about the remote host :
Operating system : Linux Kernel 2.6
NetBIOS name : (unknown)
DNS name : (unknown)

[^] Back to 192.168.1.142

Port domain (53/udp)

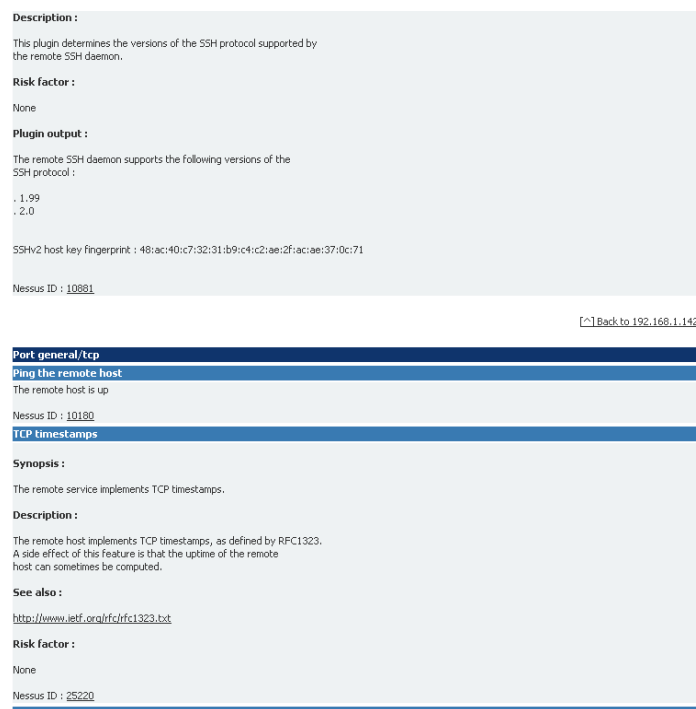
DNS Server Detection

A DNS server is running on this port. If you do not use it, disable it.

Risk factor : Low

Nessus ID : 11002

图：查看导出的 Nessus 报告具体主机信息



图：查看 Nessus 报告中检测到的端口服务信息，关注 Risk factor :内容

根据报告的内容关键字 **Risk factor** :确定是否需要安装相关的补丁程序；建议在安装补丁程序后再做一次扫描.这样前后对比一下就可以确认是否有漏掉的补丁没有打上。

定期进行弱点扫描；分析弱点扫描分析报告并修正存在的漏洞，通过套件进行更新、修补漏洞，并再次进行弱点扫描，确认漏洞是否已经获得修补。

作者简介：CU 网友 kns1024wh，目前从事 Linux 群集方面的具体工作，之前做过多年的 IT 技术支持、MCT 讲师、及 REDFLAG 的技术合作，技术专长群集、unix 主机、AD 部署等，您可以通过电子邮件 lvsheat@qq.com 或者 Chinaunix 社区与他取得联系。

具有负载均衡功能的 MySQL 服务器集群部署及实现

ChinaUnix 网友：徐连祥

摘要：MySQL 是一个高速度、高性能、多线程的关系型数据库管理系统，适用平台多，可扩展性强。在实际生产环境中，部署和实现具有一定负载均衡功能的 MySQL 服务器集群，对于提高用户数据库应用系统的性能、速度和稳定性具有明显的作用。本文简要介绍了在 FreeBSD 7.0-Release 系统上部署实现 MySQL 服务器集群的方案，并对可能出现的问题提供了相应的解决方法。

1. 引言

MySQL 是一个高速度、高性能、多线程、开放源代码，建立在客户/服务器(Client /Server)结构上的关系型数据库管理系统(RDBMS)。它始于 1979 年，最初是 Michael Widenius 为瑞典 TcX 公司创建的 UNIREG 数据库系统，当时的 UNIREG 没有 SQL(Structured Query Language 结构化查询语言)接口，限制了它的应用。1996 年 5 月，Widenius 开发出了 MySQL 的最初版本，开始在 Internet 上公开发行。MySQL 的开发人员从一开始就一直关注它的性能，为此不惜特性集，直到今天，MySQL 依然保持本色，以高速度高性能为首要原则。随着时间的推移，MySQL 也加入了大型

数据库产品的高级特性，如存储过程、视图、触发器等，使其在企业级数据库系统中开始被部署应用[1]。2008年10月，SUN公司收购了MySQL AB公司，开始进入开源领域。随着重量级操作系统Solaris的开源，SUN MySQL在数据库市场占有的份额将会进一步提高。因此，在生产环境中部署具有负载均衡功能的MySQL服务器集群，对于提高企业数据库应用系统的速度、稳定性及可伸缩性具有很大的现实意义，也可以有效降低应用系统的投资成本。本文将以前FreeBSD 7.0-Release操作系统为例，利用MySQL数据库的复制(Replication)特性，简要介绍部署MySQL服务器集群的实现方法和相关注意事项。

2. 系统模型

本集群的结构为一个主MySQL服务器(Master)服务器与多个从属MySQL服务器(Slave)建立复制(replication)连接，主服务器与从属服务器实现一定程度上的数据同步，多个从属服务器存储相同的数据副本，实现数据冗余，提供容错功能。部署开发应用系统时，对数据库操作代码进行优化，将写操作(如UPDATE、INSERT)定向到主服务器，把大量的查询操作(SELECT)定向到从属服务器，实现集群的负载均衡功能。如果主服务器发生故障，从属服务器将转换角色成为主服务器，使应用系统为终端用户提供不间断的网络服务;主服务器恢复运行后，将其转换为从属服务器，存储数据库副本，继续对终端用户提供数据查询检索服务。

3. 部署实现

本文以一台主服务器带三台从属服务器为例，简要介绍MySQL服务器集群的实现方案和具体方法步骤。

3.1 系统部署

由于FreeBSD系统对机器硬件要求较低，出于降低系统部署成本考虑，主服务器和从属服务器操作系统均采用FreeBSD 7.0-Release，并采用最小化定制安装，完成以后系统占用磁盘空间仅为254M(不计swap分区所占空间，它随具体机器内存容量的变化而变化)。为充分发挥系统硬件性能，MySQL数据库采用源代码编译安装。

3.1.1 安装FreeBSD系统

在主服务器和从属服务器上安装FreeBSD 7.0-Release，具体安装方法步骤不是本文主要内容，在此略过，如有疑问可以参考FreeBSD系统手册。需要说明的是为方便用户系统运行期间的维护管理，要打开系统的SSH服务功能，系统安装配置期间允许root用户远程登录，正常运行以后要关闭root用户的远程登录功能。在/etc/ssh/sshd_config配置文件中对PermitRootLogin设置为yes为允许root用户远程登录到系统，设置为no即为不允许。修改保存文件后执行kill -HUP `cat /var/run/sshd.pid`命令即可重启sshd守护进程，使设置生效[2]。上述操作需要root用户权限。

3.1.2 编译安装MySQL

到MySQL的官方网站<http://www.mysql.com>下载MySQL数据库的安装源代码压缩包，本例下载的是mysql-5.1.30.tar.gz，以root用户身份登录到系统，开始安装MySQL数据库系统。

由于在对MySQL进行源代码编译安装时要求使用GNU的C编译器，而FreeBSD系统本身提供的不是该编译器，因此用户必须下载安装GNU C编译器[3]。把下载的gnumake-3.81.tar.gz解压安装，按如下步骤即可安装GNU C编译器：

```
[root@FreebsdMaster/softwares]# tar -zxvf gnumake-3.81.tar.gz
```



```
[root@FreebsdMaster /softwares]# cd make-3.81
```

```
[root@FreebsdMaster /softwares/make-3.81]# ./configure ; make; make install; make clean
```

GNU C 编译器默认安装到/usr/local/bin，在安装 MySQL 时按绝对路径调用 make 命令即可，如# /usr/local/bin/make。

MySQL 的编译安装比较复杂，具体操作及相关注意事项如下：

(1). 解压 mysql-5.1.30.tar.gz，并进行配置

```
[root@FreebsdMaster /softwares]# tar -zxvf mysql-5.1.30.tar.gz
```

```
[root@FreebsdMaster /softwares]# cd mysql-5.1.30
```

```
[root@FreebsdMaster /softwares/mysql-5.1.30]# ./configure \
```

```
--prefix=/app/mysql5 --with-charset=gb2312
```

说明：MySQL 默认的安装目录是/usr/local/mysql，为了增强系统部署的灵活性，紧贴用户应用系统实际情况，可以用--prefix 参数定制安装目录。--with-charset 参数是使 MySQL 数据库支持中文 gb2312 字符集，如果需要在支持其它字符集，使用--with-extra-charset 参数，格式为--with-extra-charset=CHARSET1,CHARSET2, ... 。

(2). 编译安装 MySQL

```
[root@FreebsdMaster /softwares/mysql-5.1.30]# /usr/local/bin/make
```

```
[root@FreebsdMaster /softwares/mysql-5.1.30]# /usr/local/bin/make install
```

```
[root@FreebsdMaster /softwares/ mysql-5.1.30]# /usr/local/bin/make clean
```

注意一定要按绝对路径调用 GNU C 编译器，否则编译过程报错退出。

(3). 为系统添加 mysql 用户组 and 用户

```
[root@FreebsdMaster /]# pw group add mysql
```

```
[root@FreebsdMaster /]# adduser
```

FreeBSD 系统没有 groupadd 命令，其添加用户组的命令是 pw，添加用户命令 adduser 按系

统提示操作即可完成。当然用 `pw user add mysql` 命令也可以完成添加用户 `mysql` 的功能，但不如 `adduser` 命令功能完善。

(4). 更改/app/mysql5 目录及文件属性

```
[root@FreebsdMaster /app]# chmod -R mysql mysql5
```

```
[root@FreebsdMaster /app]# chgrp -R mysql mysql5
```

说明：/app/mysql5 为 MySQL 数据库系统所在目录，如果用户部署的应用系统数据量极大，可以将数据库系统目录设在大容量磁盘阵列上。磁盘阵列设置与具体机器硬件相关性很大，具体操作设置参考具体硬件系统说明和 FreeBSD 操作手册。不借助第三方软件，FreeBSD 7.0-Release 系统可以配置 RAID0、RAID1 磁盘阵列。

(5). 安装 MySQL 系统初始数据库

```
[root@FreebsdMaster /softwares/ mysql-5.1.30]# ./scripts/mysql_install_db.sh  
--basedir=/app/mysql5 --data-dir=/app/mysql5/data
```

注意：必须考虑 Unix 系统的文件权限特性，首先给 `mysql_install_db.sh` 脚本添加可执行属性，否则安装脚本无法执行。执行 `chmod +x ./scripts/mysql_install_db.sh` 命令即可。参数 `--basedir=/app/mysql5` 和 `--datadir=/app/mysql5/data` 为必加参数，否则脚本运行报错退出，无法安装系统初始数据库 `mysql` 和测试数据库 `test`，`mysql` 数据库包含系统权限设置表，没有这些表数据库服务进程 `mysqld` 将无法启动。

(6). 再次更改/app/mysql5 目录及其文件属性

```
[root@FreebsdMaster /app]# chmod -R root mysql5
```

```
[root@FreebsdMaster /app]# chmod -R mysql mysql5/data
```

这样做是为了保证系统安全，防止通过 MySQL 系统漏洞的入侵行为。MySQL 以 `mysql` 用户身份运行，对 `/data` 目录具有完全访问权限，对其它目录则只有系统管理员根据需要设置的访问权限，确保 FreeBSD 系统的安全。这里需要特别提醒用户注意的是要确保 `mysqld` 文件(MySQL 的守护进程，位于 `/app/mysql5/libexec` 目录下)没有设置 `setuid` 标志位，如果设置了该标志位，入侵者对其它文件就具有其不应有的权限。具体原因参考关于 Unix 系统安全的手册，本文对这个问题不再赘述。

(7). 启动 MySQL 服务器

上述操作全部完成以后，输入如下命令启动 MySQL 数据库服务守护进程：

```
[root@FreebsdMaster /]# /app/mysql5/bin/mysqld_safe --user=mysql &
```

`--user` 参数表示以 `mysql` 用户身份运行 MySQL 数据库服务守护进程，“&”符号表示以后台方

式运行。输入 `netstat -an | grep 3306` 命令，如果看到下面的输出则表示 MySQL5.1.30 已在用户系统上编译安装成功。其中，3306 是 MySQL 数据库系统默认的监听端口号。

```
[root@FreebsdMaster /]# netstat -an | grep 3306  
tcp4 0 0 *.3306 *.* LISTEN
```

按照同样方法与步骤在其它三台机器上安装 FreeBSD 7.0-Release 与 MySQL5.1.30，为下面的 MySQL 服务器集群配置准备好系统平台基础。

3.2 系统配置

所有系统安装完毕之后，需要对它们做一些必要的配置方可按照用户预先设计的结构模型实现具有负载均衡功能的 MySQL 服务器集群。

3.2.1 FreeBSD 操作系统配置

(1). 调整内核参数，优化操作系统性能

因为是处于生产环境的服务器，投入正常运行之前一定要对内核参数进行必要的调整，提升系统性能，加强稳定性。简要说就是去掉不必要的硬件支持选项，使新内核占用更少的内存，加快启动速度。不同机器硬件系统差异很大，用户可根据具体系统的硬件情况不断调整试验，最终定制出最优化、最安全的 FreeBSD 系统。一定要屏蔽掉组合键 `Ctrl+Alt+Del` 的热重启功能，防止用户误按 `Ctrl+Alt+Del` 重启系统，引起数据库系统服务意外中断。在内核配置文件如 `MYKERNEL` 中加入：

```
options SC_DISABLE_REBOOT
```

重新编译系统内核，重启后即可屏蔽 `Ctrl+Alt+Del` 组合键的热重启功能。如何定制新内核及重新编译 FreeBSD 系统，用于可参考 FreeBSD 系统手册，此不赘述。

(2). 进行其它设置，增强系统安全性

用户可根据以下一般安全准则设置自己的 FreeBSD 系统。

- 选择一个好的密码并予以妥善保护。
- 对于敏感文件采用适当安全等级的加密算法进行加密。对于敏感级别不是很高的文件采用系统自带的 `crypt` 工具加密；对于敏感度极高的文件则使用专用加密工具，如 PGP、GPG 进行不对称加密，提高加密文件的防攻击能力。
- 仔细设置文件访问权限，达到保护文件目的。
- 保护好自己的 `.profile` 文件。恰当设置 `.profile` 文件的访问权限，确保只有自己能访问。
- 仔细维护已设置 `setuid` 或 `setgid` 标志位的任何程序。
- 禁止不加防范地离开已注册终端。任何时候都必须按常规正常退出系统，先执行 `clear` 清屏命令，防止其它用户看到操作痕迹，再执行 `exit` 命令退出终端。
- 防范特洛伊木马程序。正确设置 `PATH` 变量的检索顺序，确保系统目录位于当前目录的前面。
- 防范病毒的入侵。
- 监控最近一次注册时间，确保没有他人盗用自己的帐号。

特别提示：对于生产环境的服务器，一定要保证服务器的物理安全，安全措施再完善的系统，

一旦被入侵者物理接触，所有的安全措施将形同虚设！

Unix 系统安全是一个很大的课题，本文只是提及了一些最肤浅的常识，具体内容用户可以参考相关书籍或网上资源。为确保 Unix 系统的配置具有高度的 安全性，可以使用 Internet 安全中心 (Center for Internet Security, CIS) 提供的安全测试工具 Benchmarks 或评估工具 Scoring Tools，检验和监控系统配置的安全性 [4]。

3.2.2 MySQL 数据库系统配置

为了能使上述系统实现一对多的主从复制(replication)和冗余机制，还需要对以上四个 MySQL 服务器进行一些必要的配置。

(1). 配置主机名和 IP 地址

本实现方案中把 MySQL 服务器 Master 主机名设为 FreebsdMaster，IP 地址配置为 192.168.1.100，其它三台 Slave 依次设为 FreebsdSlave1，192.168.1.101；FreebsdSlave2，192.168.1.102；FreebsdSlave3，IP 地址 192.168.1.103。

(2). 设置 root 用户密码

MySQL 的 root 用户初始密码为空，为保证系统安全必须为 root 用户设置密码：

```
[root@FreebsdMaster /]# /app/mysql5/bin/mysql -uroot -p
```

Enter password:

```
mysql>SET PASSWORD=PASSWORD( 'secret' );
```

同 Unix 系统一样，输入密码时屏幕没有任何回显，进入系统后执行 SET PASSWORD SQL 语句设置或修改 MySQL root 用户密码。为使应用系统服务器远程连接到 MySQL 数据库服务器，除本地 root 用户以外，还要设置一个能从用户内部网络其它终端登录的 root 用户。具体操作如下：

```
mysql>USE mysql;
```

```
mysql>UPDATE user SET Host=' %' WHERE Host=' hostname.yourdomain' AND  
User=' root' ;
```

```
mysql>flush privileges;
```

完成上述操作后应用系统服务器就可以连接到 MySQL 数据库服务器进行各种操作了，这个远程 root 用户的初始密码为空，用户需要从远程终端登录，为其设置密码。

(3). 开启 MySQL 服务器的二进制更新日志功能

上述 MySQL 服务器安装完成后，安装程序没有生成 my.cnf 配置文件，为此用户需要手工添加 /etc/my.cnf 文件。MySQL 服务是由 /app/mysql5/libexec/mysqld 提供的，其寻找 my.cnf 配置文件的默认顺序为 /etc/my.cnf，~/my.cnf， /usr/local/mysql/etc/my.cnf。启动 ee 编辑器，# ee /etc/my.cnf，在 Master 机器的配置文件写入如下内容：

```
[mysqld]  
log-bin=mysql-master-bin  
server-id=1
```

在其它三台 Slave 机器的配置文件 /etc/my.cnf 文件写入如下内容：


```
[mysqld]
```

```
server-id=int
```

其中，int 为一大于 1 且互不相等的正整数，如 2，3，4 等。务必要正确设置每个机器上的/etc/my.cnf 配置文件，尤其是 server-id 不能有重复，这是一对多数据库复制能否成功的关键[5]!

设定好上述/etc/my.cnf 文件后启动 Master 上的 MySQL 数据库服务时就自动开启了二进制更新日志功能。在 Slave 上启动 MySQL 时要加上—log-bin 参数，即可开启二进制更新日志功能。在 Slave 机器上启动 MySQL 服务的命令如下：

```
# /app/mysql5/bin/mysqld_safe --user=mysql --log-bin &
```

这样做的目的为了实现冗余容错功能。当主服务器 Master 出现故障停止服务时，在预转换角色成为 Master 的 Slave 上执行 STOP SLAVE; RESET MASTER SQL 语句，在其它两台 Slave 上执行 CHANGE MASTER TO SQL 语句，执行 STOP SLAVE; RESET MASTER SQL 语句的 Slave 服务器即转换成新的 Master，其余两台机器则从新的 Master 复制数据库数据(复制的是更新 UPDATE、INSERT 等操作，并不是简单的 copy)，从而实现冗余容错。

3.3 系统实现

完成上述准备后可以开始实现 MySQL 数据库的一对多复制，对应用系统的数据库访问代码进行优化，使更新操作 UPDATE、INSERT 等 SQL 语句定向到 Master 服务器，查询检索 SELECT 语句定向到 Slave 服务器，从而实现负载均衡；当主服务器 Master 出现故障停止服务时，通过服务器角色转换实现冗余容错；上述所有服务器通过高速核心交换机连接在一起，协同工作，提供集群 (Cluster) 性能。下面，本文就此举一反三，具体说明实现(Implementation)步骤。

3.3.1 准备主服务器

(1). 在主服务器上创建示例数据库 repl_db

从本地或远程终端登录到 Master 服务器，执行 CREATE DATABASE repl_db;语句，创建示例数据库 repl_db。建立表 repl_table：

```
mysql>CREATE TABLE repl_table (f1 INT, f2 VARCHAR(20)) ;
```

向表中添加数据：

```
mysql>INSERT INTO repl_table (f1, f2) VALUES(1, 'first' );
```

(2). 授予 Slave 复制(Replication)权限

授予从属服务器 Slave 进行复制的权限。假设从属服务器连接到主服务器复制的用户名为“repl”，密码为“g00r002b”：

```
mysql>GRANT REPLICATION SLAVE ON *.* TO repl@'%' IDENTIFIED BY 'g00r002b' ;
```

刷新系统权限表，锁定写入语句操作：

```
mysql>FLUSH TABLES WITH READ LOCK;
```

这时所有写入操作都被锁定，包括支持事务(Transaction)特性的 InnoDB 类型表的提交 (COMMIT)操作也被锁定，为数据库初始复制(copy)做好准备。

(3). 简单备份 repl_db

在此，通过 tar 命令把数据库初始数据备份到/tmp 目录。

```
# tar -cvf /tmp/mysql-data.tar /app/mysql5/data/repl_db
```

(4). 记录主服务器状态

执行 SHOW MASTER STATUS 语句：

```
mysql> SHOW MASTER STATUS;
```

```
-----+-----+-----+-----+
| File | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-master-bin.000001 | 1027 | | |
+-----+-----+-----+-----+
```

记下 File 和 Position 两个参数的值，从属服务器 Slave 为了复制(replication)连接到主服务器 Master 时要用到这两个参数，如果参数与此不符将导致复制(replication)失败！

(5). 释放主服务器表级写入锁定

```
mysql> UNLOCK TABLES;
```

至此，主服务器解除写入锁定，一对多复制准备工作已全部完成，准备接受 Slave 的复制(replication)连接。

3.3.2 准备从属服务器

(1). 停止从属服务器的 MySQL 数据库服务

```
# /app/mysql5/bin/mysqladmin -u root -p shutdown
```

Enter password:

输入 MySQL 系统 root 用户密码，MySQL 服务停止。

(2). 简单 copy 数据库初始数据

在从属服务器上执行 ftp 操作，访问主服务器，下载/tmp/mysql-data.tar 文件，在/app/mysql5/data 目录下执行如下操作，把初始数据 copy 到 Slave 服务器：

```
[root@FreebsdSlave1 /app/mysql5/data]# tar -xvf /tmp/msyql-data.tar
```

其它两台 Slave 服务器执行同样操作，实现数据库初始数据的简单 copy。

(3). 重启从属服务器上的 MySQL 数据库服务

```
# /app/mysql5/bin/mysqld_safe --user=mysql --log-bin &
```

从属服务器已启动，同时启动二进制更新日志功能，为角色转换做好准备。

(4). 登录到从属服务器 Slave 的 MySQL 客户端

```
# /app/mysql/bin/msyql -uroot -p
```

(5). 启动从属服务器上的复制(replication)线程

```
mysql> CHANGE MASTER TO
```

```
-> MASTER_HOST='192.168.1.100',
```

```
-> MASTER_USER='repl',
```

```
-> MASTER_PASSWORD='g00r002b',
```

```
-> MASTER_LOG_FILE='mysql-master-bin.000001',
```

```
-> MASTER_LOG_POS=1027;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> START SLAVE;
```

```
Query OK, 0 rows affected (0.00 sec)
```

在所有从属服务器上执行相同的操作，自此从属服务器已连接到主服务器，开始真正意义上的 replication 工作。测试从属服务器的复制工作是否正常：

```
mysql> show slave status\G
```

```
***** 1. row *****
```

```
Slave_IO_State: Waiting for master to send event
```

```
Master_Host: 192.168.1.100
```

```
Master_User: repl
```

```
Master_Port: 3306
```

```
Connect_Retry: 60
```

```
Master_Log_File: mysql-master-bin.000001
```

```
Read_Master_Log_Pos: 1027
```

```
Relay_Log_File: FreeBSDSlave1-relay-bin.000005
```

```
Relay_Log_Pos: 251
```

```
Relay_Master_Log_File: mysql-master-bin.000001
```

```
Slave_IO_Running: Yes
```

```
Slave_SQL_Running: Yes
```

看到上述信息说明从属服务器已启动了与 replication 相关的线程 I/O 和 SQL，一对多的 replication 已经开始工作。

(6). 功能测试

在主服务器上写入新的数据:

```
mysql>INSERT INTO repl_table (f1, f2) VALUES(2, 'second' );
```

```
mysql>SELECT * FROM repl_table;
```

```
+-----+-----+
```

```
| f1 | f2 |
```

```
+-----+-----+
```

```
| 1 | first |
```

```
| 2 | second |
```

```
+-----+-----+
```

2 rows in set (0.01 sec)

在 slave 上执行相同的查询操作:

```
mysql>SELECT * FROM repl_table;
```

```
+-----+-----+
```

```
| f1 | f2 |
```

```
+-----+-----+
```

```
| 1 | first |
```

```
| 2 | second |
```

```
+-----+-----+
```

2 rows in set (0.01 sec)

得到相同的查询结果,说明复制(replication)机制已成功开始工作!

3.3.3 配置网络 DNS 服务器

在 BIND DNS 服务器中,为上述三台从属服务器 Slave 配置同一个名字,客户端的查询检索操作将由 DNS 服务器定向到其中的一台 Slave。因此,对于同一名字,不同的客户端会定向到不同的地址,访问不同的 MySQL 服务器,从而达到负载均衡的目的。假设用户为三台 Slave 分配的 DNS 名字为 mysqlslave.yourdomain, DNS 服务器区域文件/var/named/yourdomain.zone 中应包含如下数据项[6]:

```
sqlmaster.yourdomain. IN A 192.168.1.100
```



```
sqlslave1.yourdomain. IN A 192.168.1.101
```

```
sqlslave2.yourdomain. IN A 192.168.1.102
```

```
sqlslave3.yourdomain. IN A 192.168.1.103
```

```
sqlslave IN CNAME sqlslave1
```

```
sqlslave IN CNAME sqlslave2
```

```
sqlslave IN CNAME sqlslave3
```

当客户端进行查询操作时，提交给主机 sqlslave.yourdomain 的请求将由 DNS 服务器随机定向到三台 Slave 中的一台，由其执行查询作业，返回结果。从而在三台 Slave 之间实现查询级别的负载均衡。

3.3.4 应用系统程序代码优化

实际应用中，对数据库的写入操作相对查询操作少得多，因此，优化应用程序的数据库连接代码，把写入操作定向到 Master 服务器，查询操作定向到 Slave 服务器，提供主服务器和从属服务器之间更新、查询的负载均衡功能。本文以 PHP 数据库连接代码为例，简要介绍代码优化方法[7]。

(1). 准备不同的数据库连接配置文件

准备两个数据库连接文件 mysql_connect_master.php 和 mysql_connect_slave.php，其文件内容如下：

mysql_connect_master.php 文件的内容：

```
// Connect to the Database Server
```

```
$linkID = @mysql_connect("sqlmaster.yourdomain", "user", "password") or die("Sorry, could not connect to the database!");
```

```
// Select the Database
```

```
@mysql_select_db("repl_db") or die("Sorry, Could not select database!");
```

```
?>
```

mysql_connect_slave.php 文件的内容：

```
// Connect to the Database Server
```

```
$linkID = @mysql_connect("sqlslave.yourdomain", "user", "password") or die("Sorry, could not connect to the database!");
```

```
// Select the Database
```

```
@mysql_select_db("repl_db") or die("Sorry, Could not select database!");
```

```
?>
```

(2). 优化程序 SQL 语句代码

对应用程序中访问数据库的代码段做出如下优化：

```
...
```

```
// $sql is the query string to be committed to MySQL server.
```

```
if ( strpos( $sql, " SELECT" ) )
```

```
{
```

```
include_once("./mysql_connect_slave.php");
```

```
...
```

```
mysql_close();
```

```
}
```

```
else
```

```
{
```

```
include_once("./mysql_connect_master.php");
```

```
...
```

```
mysql_close();
```

```
}
```

...

?>

说明：每当向 MySQL 服务器提交数据库操作时，加入一个 if-else 判断语句对变量 \$sql 进行判断定向，如果是 SELECT 查询操作，将其定向到从属服务器 Slave 中的一个；如果是其它更新语句，则将其定向到主服务器 Master。此处代码段是应用系统实现负载均衡的关键，用户务必根据自己实际情况，写出准确无误的代码。每次对数据库操作完毕都要及时释放数据库连接，以免更新、查询操作分别定向失败，影响系统负载均衡功能。这样做虽然在一定程度上增加 Web 服务器或应用服务器的开销，但与 MySQL 服务器集群负载均衡功能带来的大幅性能提升及冗余容错特性相比，这个开销绝对是物超所值！

3.4 注意事项

正确部署实现具有负载均衡功能的 MySQL 服务器集群必须注意以下事项：

(1). MySQL 数据库复制(replication)特性是核心

此处的复制不是简单的 copy，从属服务器启动两个线程(thread)：I/O 线程和 SQL 线程，I/O 线程接收主服务器对参与复制数据库的更新操作事件(event)，并记入自己的中继二进制更新日志文件(hostname-relay-bin.00000n)，由 SQL 线程将更新操作写入自己的数据库表项。主从服务器之间复制的不是具体的数据内容，而是具体的以二进制格式记录的操作事件，因而在一定程度上实现主从服务器之间的数据同步。(这种复制类似于生物学意义上的按基因复制，在英语中 replication 的主要词义就是指该种复制。)

(2). 复制的复杂性

主从数据库服务器间的 replication 要求 Master 与 Slave 上的 MySQL 版本最好一致，主从服务器必须设置相同的字符集，否则很容易造成复制失败。主服务器上更新权限表内容的 FLUSH 语句不会被复制[8]。

(3). 按照范式化要求设计数据库

生产环境下基于 MySQL 服务器的应用系统要想稳定运行，按范式化设计系统数据库是基本要求，具体内容可参考相关书籍。

(4). 打开数据库服务器的远程用户连接功能

打开主从服务器的远程用户连接是实现更新、查询操作分别定向的必要条件，否则，来自应用服务器的连接请求失败，影响系统应用正常运行。

(5). 负载均衡功能的实现需要良好的团队合作

BIND DNS 服务器实现了从属服务器 Slave 之间的负载均衡，Slave 和 Master 之间的负载均衡则由应用系统开发人员在程序代码级实现。整个系统的性能提升和冗余容错需要网络管理和应用系统开发团队之间的良好合作，否则负载均衡功能的实现就会失败。

4. 常见问题

(1). 如何估算 MySQL 服务器集群的性能提升量？

针对本文采用的结构模式，可对应用系统整体性能提升做出大致估算。假设应用系统写操作占 10%，读操作占 90%，写操作耗时是读操作的 2 倍，系统的吞吐量(throughput)为 T(用 reads/s 读操作次数/秒来衡量)。把写操作线性转换为读操作，则有：

$T = 2X_{\text{writes}} + 9X_{\text{writes}} \Rightarrow \text{writes} = T/11$ ① (不采用主从复制模式，读写操作集中到一个服务器上)

$T = 2X_{\text{writes}} + 9X_{\text{writes}}/N \Rightarrow \text{writes} = T/(2+9/N)$ ② (采用一对多的主从复制模式，读操作在从属服务器，写操作在主服务器)

其中，writes 为系统单位时间内所能承受的最大写操作次数，N 为从属服务器个数，N 大于等于 2。在不采用主从复制模式时，系统性能 $\text{writes} = T/11$ ；采用本文一对三的复制模式时，系统性能 $\text{writes} = T/5$ 。采用负载均衡模式与不采用系统性能之比为 11:5，即 2.2:1，考虑到应用服务器的额外开销，系统整体性能提升了整整 1 倍！从②式可以看出，系统整体性能理论极限为 T/2，当然在实际生产环境中不可能达到。具体部署时用户可以根据自己的实际情况估算出合理的从属服务器数量，主要影响因素是网络带宽和机器整体性能[9]。

(2). 如何应对主从服务器崩溃？

当某台从属服务器崩溃时，修复故障重启后重新连接到主服务器，根据其 master.info 文件更新其数据，保持与主服务器的数据同步。如果主服务器崩溃，在某一从属服务器上执行 STOP SLAVES; GRANT REPLICATION SLAVE ON *.* repl_db TO 'repl' '@' '%' IDENTIFIED BY 'g00r002b'; RESET MASTER; 这三个 SQL 语句，由于从属服务器已启动了二进制更新日志功能，因此具备了角色转换的必要条件。更改其主机名、IP 地址及 server-id 与 Master 一样，重启 MySQL 服务器，系统开始正常对外提供服务。其它两台从属服务器则不需执行任何操作，继续执行 replication 过程。BIND DNS 服务器和应用程序也不需做任何调整，继续对系统用户提供不间断服务。主服务器排除故障恢复正常后，将其网络配置改为与现有 Master 转换角色之前一样的配置，重启 MySQL 服务，将其角色转换为从属服务器。也就是说，整个集群机器的角色可以相互循环转换，提高系统的冗余性和可靠性。在此需要注意的是，在应用系统调试运行正常之后，在 Slave 角色服务器的/etc/my.cnf 文件[mysqld]段加入 slave-skip-errors=all，保证集群之间复制(replication)的正常运行。

5. 结束语

部署与实现具有负载均衡功能的 MySQL 服务器集群是一项复杂的系统工程，需要多方面良好的协同合作才能做好。服务器的搭建配置、BIND DNS 服务器的配置部署，以及应用系统程序的开发都要紧紧围绕实现 MySQL 服务器集群负载均衡功能这个目标。必须对主服务器的运行状态进行动态监控，如果发生故障，立即执行角色转换过程，确保为终端用户提供可靠、不间断的服务。可以针对具体系统环境写出监控脚本或程序，确保系统的可靠性与稳定性。

参考文献

- [1] W.Jason Gilmore. Beginning PHP and MySQL5 From novice to Professional(Second Edition) 573-580
- [2] <http://www.freebsd.org/>
- [3] <http://www.mysql.com/>
- [4] Kenneth Rosen. UNIX:The Complete Reference (Second Edition) 273, 274
- [5] Ian Gilfillan. MySQL4: 从入门到精通 301-323
- [6] 林慧琛，刘殊，尤国君. Red Hat Linux 服务器配置与应用 88-92

[7] W.Jason Gilmore. Beginning PHP and MySQL5 From novice to Professional(Second Edition) 161, 509-511

[8] Ian Gilfillan. MySQL4: 从入门到精通 301-323

[9] <http://www.mysql.com/>

红帽 RHEL5U3 平台实现 pppoe 接入记账认证服务器

ChinaUnix 网友: zorro

软件需求:

红帽自带的: rp-pppoe,ppp,freeradius,freeradius-mysql,mysql。

另需: freeradius-2.1.3 以及 ppp2.4.4 的源代码包。

第一部分: 实现简单的 pppoe 拨号服务

检查系统上的 rp-pppoe 以及 ppp 软件是否安装, 命令:

```
#rpm -q rp-pppoe ppp
```

如果安装则会有显示软件包的名称, 若未安装, 找到安装盘上的相关 rpm 包安装。

然后配置/etc/ppp/pppoe-server-options 文件, 内容为:

```
# PPP options for the PPPoE server
# LIC: GPL
require-chap
login
lcp-echo-interval 10
lcp-echo-failure 2
defaultroute
ms-dns 192.168.1.1
logfile /var/log/pppod.log
```

文件内容不做解释, 可以通过 man pppd 查寻相关帮助。此时 pppoe 服务器从/etc/ppp/chap-secrets 文件中读取验证的用户名和密码信息, 此文件格式为:

```
# Secrets for authentication using CHAP
# client      server      secret          IP addresses
"zorro"      *          "123456"        *
```

想要多添加用户, 再加入新行即可。

然后可以启动 pppoe 服务了, 命令是:

```
# pppoe-server -l eth0 -l $LOCALIP -R $REMOTEIP -N $CONNLIMIT
```

命令详细信息查询 man pppoe-server。

注: 红帽自带的 pppoe 服务器, 在 RHEL5 至 RHEL5U2 的版本上与系统的 syslog 服务有冲突。可现关闭 syslog 服务, 让 pppoe 服务器正常运行。

第二部分：让 pppoe 服务使用 freeradius 服务器进行验证

首先更改/etc/ppp/pppoe-server-options，添加一行设置：

```
# PPP options for the PPPoE server
# LIC: GPL
require-chap
login
lcp-echo-interval 10
lcp-echo-failure 2
defaultroute
ms-dns 192.168.1.1
logfile /var/log/pppod.log
plugin /usr/lib/pppd/2.4.4/radius.so
```

表示让 pppoe-server 在运行时加入 pppd 的 radius 查检，可以让 pppoe 服务通过 freeradius 来验证和记账。添加此行之 后，可以创建/etc/radiusclient/目录，创建完之后，我们需要使用一下 ppp 的源代码，因为红帽没有给我们提供相应的配置文件。解压 ppp 的源代码目录：

```
# tar xvfz ppp-2.4.4.tar.gz
# cd ppp-2.4.4
# cd pppd/plugins/radius/etc/
# cp * /etc/radiusclient/
```

这个目录下我们需要的所有相关的配置文件，其中最重要的是/etc/radiusclient/radiusclient.conf，我们先打开此文件配置，其内容为（已去掉注释）：

```
auth_order    radius
login_tries    4
login_timeout  60
nologin /etc/nologin
issue    /etc/radiusclient/issue
authserver    localhost:1812
acctserver    localhost:1813
servers       /etc/radiusclient/servers
dictionary    /etc/radiusclient/dictionary
login_radius  /sbin/login.radius
seqfile       /var/run/radius.seq
mapfile       /etc/radiusclient/port-id-map
default_realm
radius_timeout    10
radius_retries    3
login_local    /bin/login
```

字段相关意义请参考文件自带的注释。

另外，还需更改此目录下的 servers 文件，此文件用来指定读取的 radius 服务器的主机名称以

及 key 值（需要在 freeradius 配置中指定）。

此时 pppoe 服务已经可以通过 radius 认证了。然后进行 freeradius 的配置。

首先检查 freeradius 是否安装：

```
# rpm -q freeradius freeradius-mysql
```

其中 freeradius-mysql 包使用来让 freeradius 连接 mysql 数据库的，本部分还用不到。

首先打开/etc/raddb/clients.conf 配置客户端访问控制，文件内容如下（已删除注释）：

```
client 127.0.0.1 {
    secret          = hello
    shortname       = localhost
    nastype        = othe
}
```

表示客户端之允许从 127.0.0.1 的 ip 登录 radius 服务，并且需要验证的 secret 为 hello，就是在上面的 servers 文件中需要配置的信息。若要实现可以从别的机器访问，请参考注释获得帮助。

然后配置 naslist 文件，内容为：

```
# NAS Name      Short Name  Type
#-----
#portmaster1.isp.com  pm1.NY    livingston
#portmaster2.isp.com  pm1.LA    livingston
localhost          local      portslave
```

此文件用来配置记录有哪些指定的 nas 服务器需要使用 radius 进行记账。现在指定的是 localhost。

主控配置文件是 radiusd.conf，目前没有需要改的，可通过注释学习其中配置方法。此文件主要是用来指定 freeradius 服务器默认的 验证和记账方式。我们目前使用本地的文件方式，就是/etc/raddb/users 文件，在其中添加我们需要提供给 pppoe 服务认证的用户信息，内容如下（某段截取）：

```
zorro Auth-Type := Local, Simultaneous-Use := 1, User-Password == "zorro"
      Service-Type = Framed-User,
      Framed-Protocol = PPP,
      Framed-IP-Address = 10.0.0.33,
      Framed-IP-Netmask = 255.255.255.255
```

其中 Simultaneous-Use := 1 字段用来设置每个用户同时登录的个数。所有字段信息都在 freeradius 的源代码中有文档给予解释。另外要注意，添加的用户字段应写在文件 DEFAULT 字段前面添加，否则可能不会生效。

都配置完毕后，可以通过 radiusd -X 命令以排错方式启动，此时再启动 pppoe-server,用客户端拨号验证一下，检查 pppoe 服务是否成功通过 freeradius 来验证用户。如果成功，这一部分完成。可通过 service radiusd restart 来正常启动 radius 服务。

第三部分：配置 freeradius 从 mysql 数据库读取用户信息

首先更改/etc/raddb/radiusd.conf,首先将\$INCLUDE \${confdir}/sql.conf 字段这行原有的注释去掉，然后找到 authorize 字段中的 sql 将注释取消掉，以及 accounting 字段中的 sql 的注释取消。然后需要更改 sql.conf 中的设置，主要是设置读取的 mysql 服务器的相关信息，可更改如下一些字段：

```
server = "localhost"           # 数据库服务器
login = "mysql_username"       # 数据库用户
password = "mysql_password"    # 数据库密码
radius_db = "radius"           # 数据库名称
```

当然前提是你现在把 mysql 服务设置好，设置方法是：

先启动你的 mysqld：

```
service mysqld start
```

然后设置 root 帐户密码：

```
mysqladmin -u root -p password 12345
```

创建 radius 数据库

```
mysqladmin -u root -p123456 create radius
```

建立 mysql 的数据库 radius 的表，我们可以通过软件已经给出的 sql 脚本导入数据库：

```
mysql -u root -p123456 radius <
```

```
/usr/share/doc/freeradius-1.1.3/examples/mysql.sql
```

这样，所有 radius 需要的数据库和表都已经建立好了。

此时可以添加测试帐号来测试是否正常工作。具体方法就不在这说了。我们待会要使用 freeradius 源代码自带的 dialup_admin 来进行数据库的管理工作，那是一个基于 web 的管理界面，使用 php 编写，要比直接用 sql 语句方便的多。

这里要注意的是，如何在数据库里添加限制没用户同时登录个数的限制，比如，我先用 sql 语句添加一个测试用户，然后来限制他的登录个数，语句是：

```
mysql -u root -p123456 radius
```

```
insert into radgroupreply (groupname,attribute,op,value) values ('user', 'Auth-Type','=','Local');
```

```
insert into radgroupreply (groupname,attribute,op,value) values ('user', 'Service-Type','=','Framed-User');
```

```
insert into radgroupreply (groupname,attribute,op,value) values ('user', 'Framed-IP-Address','=','255.255.255.254');
```

```
insert into radgroupreply (groupname,attribute,op,value) values ('user', 'Framed-IP-
```

```
Netmask',':=','255.255.255.0');
```

```
insert into radcheck (username,attribute,op,value) values ('test','User-Password',':=','test');
```

```
insert into usergroup (username,groupname) values ('test','user');
```

这样就给我们的数据库中添加了一个帐户测试帐户 test，并让 test 帐户属于 user 组。

然后我们可以通过设置 user 组的检查属性来设置此组帐户的登录数限制，以后只要用户加入这个组，登录数都不能超过 1 次：

```
INSERT INTO radgroupcheck (GroupName, Attribute, op, Value) values("user",  
"Simultaneous-Use", ":", "1");
```

好了，限制完成。我们现在可以使用 test 帐户登录，并测试是否是通过 mysql 数据库验证的。并且可以检查是不是只能登录一个。

第四部分：使用 dialup_admin 来管理 radius 服务器

服务器搭建成功后如果需要 web 管理，freeradius 自带了一个 web 管理程序 dialup_admin。

首先测试 Apache 与 PHP 安装的正确性。Apache 的 WEB 目录在 /var/www/html 下首先测试 Apache，在浏览器输入 Http://127.0.0.1 下看到 Apache 界面表示正常。

编辑一个 PHP 测试文件，内容为 ，保存为 phpinfo.php。在浏览器输入 Http://127.0.0.1/phpinfo.php 看到 php 相关信息页面，表示 php 运作正常。

直接将 freeradius 源代码中的 dialup_admin/文件夹全部拷贝到 Apache 网页目录下/etc/local 下。

修改 dialup_admin/conf/下的 admin.conf 文件中的如下参数：

```
sql_type:mysql  
sql_server:localhost          //sql 地址  
sql_port:3306                 //默认端口  
sql_username:root  
sql_password:123456           //密码  
sql_database:radius           //数据库名
```

```
general_base_dir: /usr/local/dialup_admin //dialup_admin 主目录  
general_domain: company.com //这个可改也可不改，没什么影响  
general_radius_server_auth_proto: chap //将 pap 改成 chap  
general_encryption_method: clear //将 crypt 改成 clear
```

说明一下，general_encryption_method: clear 是必须得改的，否则用 web 管理界面添加用户后会出现密码被加密后无法识别导致认证失败，clear 方式是以明文存储密码，freeradius 默认是用明文存储密码，两者要对应，所以这里要用 clear 方式。

修改 dialup_admin/conf/下的 naslist.conf 文件如下


```
nas1_name: nas1. %{general_domain}
nas1_model: Computer           //NAS 服务器的类型
nas1_ip: 192.168.1.1          //NAS 服务器的 IP 地址
nas1_port_num: 15
nas1_community: public
```

其他的通通删掉或用#号注释掉，这样修改的目的是方便在 web 管理界面中直观的看到每个 NAS 服务器连接用户的情况。

导入相关数据库表

```
cd /usr/local/dialup_admin/sql
mysql -uroot -p123456 radius < userinfo.sql
mysql -uroot -p123456 radius < totacct.sql
mysql -uroot -p123456 radius < mtotacct.sql
mysql -uroot -p123456 radius < badusers.sql
```

dialup_admin 用的是 php3 格式的文件，需要 web 服务器支持这种格式，修改/etc/httpd/conf/httpd.conf 文件，在此文件最后添加：

```
AddType application/x-httpd-php .php .html .htm .php3
```

并在默认的主页目录下建立一个指向/usr/local/dialup_admin/htdocs 的符号连接 htdocs
保存后重启 httpd 服务

```
service httpd restart
```

访问：<http://localhost/htdocs/index.html>

能正常打开各页面就 ok 了。默认是打开的 sql 调试模式。可以在 admin.conf 中找到 sql_debug.把 true 改为 false 即可关闭。

至此，整个 Radius 服务器搭建完成。

根文件系统所在 LVM 无损收缩全攻略

ChinaUnix 网友：Jerrywjl

作为在几乎所有 Linux 发行版操作系统中所带的逻辑卷管理方式（LVM），其最大的特点是部署灵活和操作方便。而且在 Red Hat Enterprise Linux 中 LVM 也一直被作为默认的磁盘管理方式直接管理系统所在的设备文件。同时 LVM 能够很好地支持例如软件 Raid，裸设备等特殊磁盘类型，以及自带包括线性、条带、镜像、快照等多种功能，甚至在高版本系统中的 Lvm2 支持 RHCS（Red Hat Cluster Suite）中的 HA LVM 和 Cluster LVM，能够实现 HA 集群中对块设备的资源共享。

但在大多数时候，恐怕一般用户使用不到其中的这些功能，倒是一些关于 LVM 的 extend 和 reduce 这方面的操作屡见不鲜，并被所有人津津乐道。

显然，从大多数操作系统角度来看，部署了 LVM 的存储设备本身也是块设备，其 extend 和 reduce 这方面的工作进行的前后，对于一般的应用势必涉及到在上面的文件系统的调整。

为此 RHEL 5 在 RHEL 4 基础上提供了 resize2fs 这种工具专门从事这种勾当。当然所谓树大招风，LVM 的名气不胫而走之后，而且在很多人的眼里，LVM 的 extend 和 reduce 结合 resize2fs 的黄金组合似乎独步天下，并在块设备和文件系统的调整中无往不利。

其实 LVM 的调整方面，被大多数发行版系统官方支持的是 lvm 的扩展，即 extend。但在尝试了这种功能的平稳性之后，大家似乎开始对 LVM 的收缩即 reduce 有更多的兴趣。因此在各种网站上使用 lvreduce 来对块设备瘦身的帖子也层出不穷，尽管官方一天到晚扯着嗓子喊这是一种不被支持的操作并且需要自负风险的操作形式。

应该说在民间其实总不乏一些富有尝试精神的高手，一边冒着在刀尖上跳舞的风险来给我们带来很多边缘的惊喜，一边抱着中了自己化骨绵掌的磁盘在寒风中痛哭流涕。当无数的前人拍死在沙滩上的同时，更富有精神的后辈不惜将前人的勇敢和变态同时发扬光大。对 LVM 的 reduce 操作不光瞄上了普通的文件系统，而且瞄上了系统的命根子——根分区。

本来不想搭理这些个人认为无端的需求：因为作为稳定要求为至高境界的生产系统和生产环境，本应该在刚刚开始部署的时候有慎重的考虑和规划，一旦部署上线之后就绝不应该或者极少对底层尤其是存储设备大动干戈。但突然发现最近这方面的需求和探求者越来越多，一方面我对大家要榨干 LVM 充分信任并要榨干其最后一滴油的想法表示理解；另外一方面正因为没有做过，所以也隐隐感觉到自己和其他人一样在心底也隐藏着这样的邪念，为此最终决定找台测试环境来实现大家梦寐以求的壮举——对部署了根文件系统的逻辑卷进行缩根操作！尽管从我个人的理论角度认为，这种操作在经过精确计算是没有问题的，但是有时候人就是比较犯贱，你告诉他不行，非要自己亲身体验一把方能找到快感。

既然如此，所以我就在这里先行爽上一把，然后把我的快感告诉大家！但是先要郑重声明，本人乃遵纪守法好公民，既然各个发行版将 LVM 收缩视为洪水猛兽，那么对于各发行版部署文档中提到的 LVM 收缩的禁区绝对言听计从，之所以在这里操作实在是为了要帮大家出个头而已。但这个操作有没有风险性，多大风险性，真的在企业中生产环境上操作可能会导致多少人被炒鱿鱼、失心疯、跳楼卧轨等等的后果本人没有时间也没有兴趣考虑，所以各位下手之前最好做足够的测试和论证，如果因为计算失误或者操作失误导致的任何问题，本人只能表示深深默哀，且爱莫能助。所以各位施主和看官在实施自己的犯罪计划之前望慎之又慎，善哉善哉！

按照本人做事的惯例，首先提供一个缩骨大法前系统的基本环境：

如图所示：RHEL5 Update 2 系统，采用默认分区方式，根分区在 LVM 上，文件系统大小为 8G：

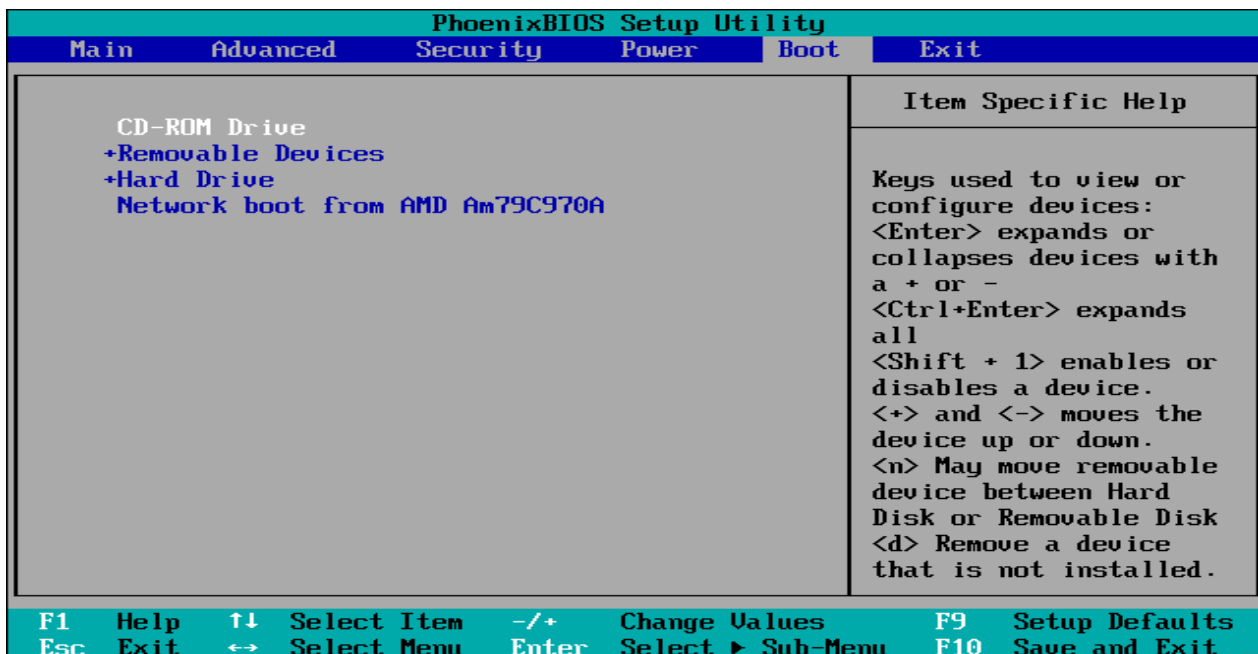
```

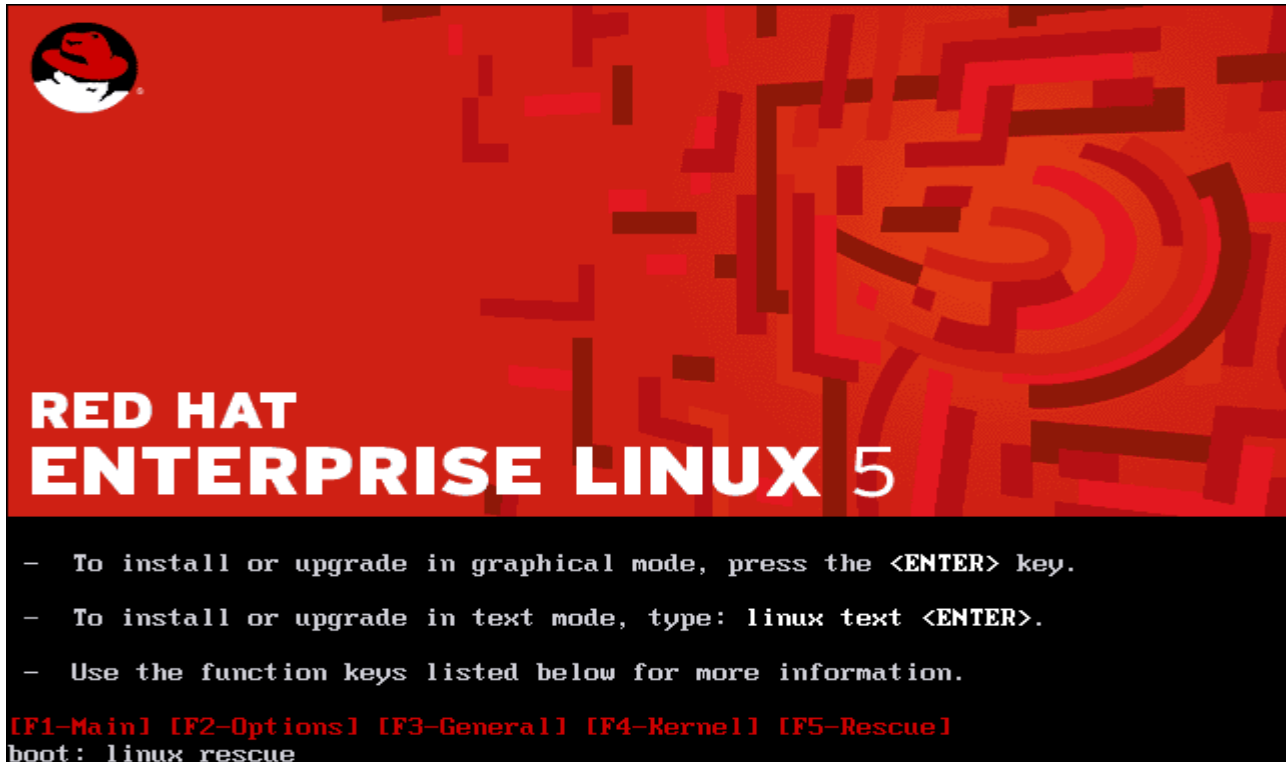
Applications  Places  System
root@dhcp-0-199:~
File Edit View Terminal Tabs Help
[root@dhcp-0-199 ~]# uname -r
2.6.18-92.el5xen
[root@dhcp-0-199 ~]# df -TH
Filesystem      Type      Size      Used     Avail  Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
ext3           7.7G      3.5G      3.9G    47% /
/dev/sda1       ext3       104M      14M      85M     14% /boot
tmpfs          tmpfs      119M       0        119M    0% /dev/shm
[root@dhcp-0-199 ~]# CD
bash: CD: command not found
[root@dhcp-0-199 ~]# cd
[root@dhcp-0-199 ~]#

```

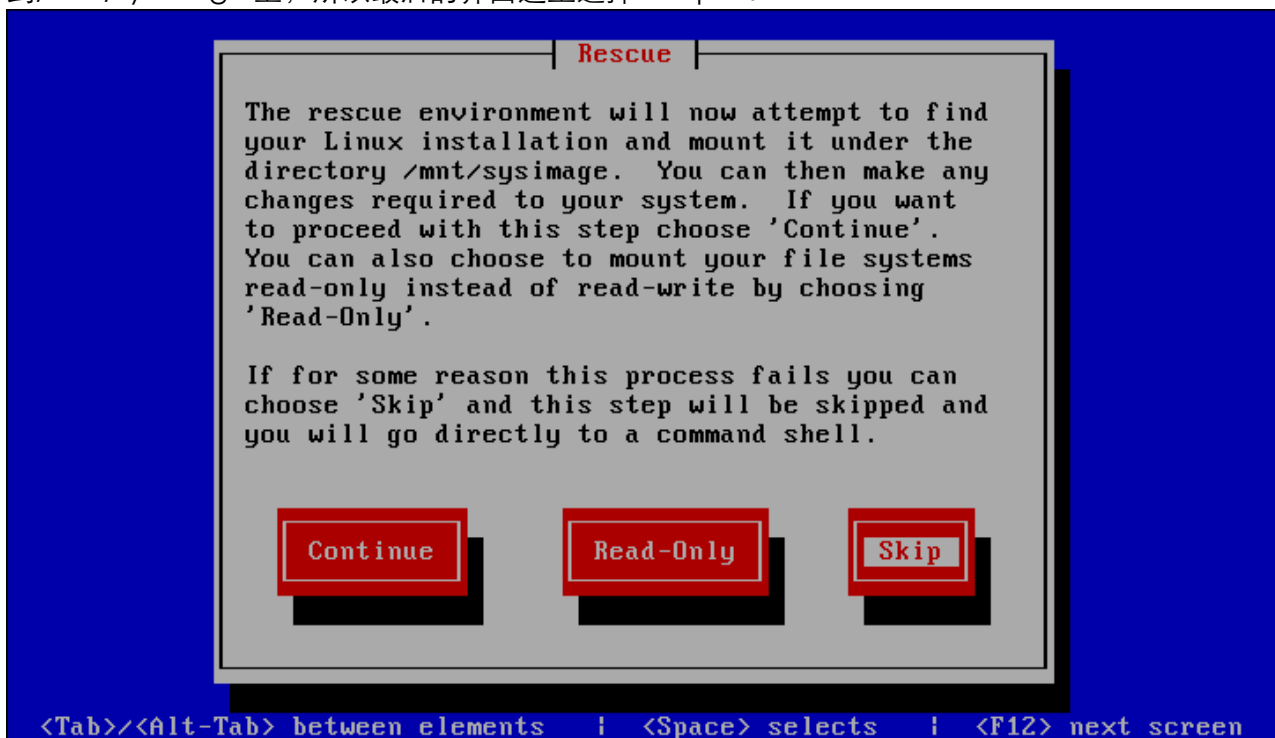
然后找张光盘启动系统执行 linux rescue 进入到 Rescue 模式：

有人曾经问我，如何在生产环境下 online 状态下来缩根，哥们，做人要厚道，事情不要做得太绝！





选择英文语言支持键盘布局，不选择网络，最后的界面是个关键，不要让系统根挂载到/mnt/sysimage 上，所以最后的界面这里选择 “skip”。



根据我上述的情况，我初步是准备计划将根分区缩小 3 个 G，但是事实上经过操作发现，文件系统在收缩之后，块设备能缩多少这往往不是根据个人意愿而定，而是要根据当前实际文件系统大小来决定块设备也就是 LVM 的边界。如果不遵循这种规律，那么你会死得很难看。

那么现在我要做的工作，首先是要探测并激活所有的 LVM 设备：

```
When finished please exit from the shell and your system will reboot.

sh-3.2# fdisk -l

Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1    *           1          13       104391   83   Linux
/dev/sda2                14        1044      8281507+  8e   Linux LVM
sh-3.2# lvm.static pvscan
/dev/hdc: open failed: Read-only file system
Attempt to close device '/dev/hdc' which is not open.
PV /dev/sda2   VG VolGroup00   lvm2 [7.88 GB / 0   free]
Total: 1 [7.88 GB] / in use: 1 [7.88 GB] / in no VG: 0 [0   ]
sh-3.2# lvm.static vgscan
Reading all physical volumes. This may take a while...
/dev/hdc: open failed: Read-only file system
Attempt to close device '/dev/hdc' which is not open.
Found volume group "VolGroup00" using metadata type lvm2
sh-3.2# lvm.static lvscan
inactive          '/dev/VolGroup00/LogVol00' [7.38 GB] inherit
inactive          '/dev/VolGroup00/LogVol01' [512.00 MB] inherit
sh-3.2# _
```

我通过 lvm.static 命令来在当前环境中加载所有 LVM 的静态库，这样即可在 rescue 模式下对逻辑卷设备进行操作。首先的工作自然是要将设备 scan 出来。然后再激活卷组：

```
sh-3.2# lvm.static vgchange -ay /dev/VolGroup00
2 logical volume(s) in volume group "VolGroup00" now active
sh-3.2# lvm.static lvscan
ACTIVE          '/dev/VolGroup00/LogVol00' [7.38 GB] inherit
ACTIVE          '/dev/VolGroup00/LogVol01' [512.00 MB] inherit
sh-3.2# resize2fs /dev/VolGroup00/LogVol01 5120M
resize2fs 1.39 (29-May-2006)
resize2fs: Bad magic number in super-block while trying to open /dev/VolGroup00/LogVol01
Couldn't find valid filesystem superblock.
sh-3.2# resize2fs /dev/VolGroup00/LogVol00 5120M
resize2fs 1.39 (29-May-2006)
Please run 'e2fsck -f /dev/VolGroup00/LogVol00' first.

sh-3.2# e2fsck -f /dev/VolGroup00/LogVol00
e2fsck 1.39 (29-May-2006)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/VolGroup00/LogVol00: 145564/1933312 files (0.4% non-contiguous), 894782/1933312 blocks
sh-3.2# _
```

大凡对 LVM 进行收缩，必须先收缩其上的文件系统。实在搞不懂原理的哥们可以这样想，文件系统相当于你的大脑，LVM 相当于你的脑壳，如果你的大脑体积比你的脑壳还要大，那么……原谅

我，我是个粗人。言归正传，在进行文件系统调整之前，系统要求对其先执行 fsck。那么如上图照做即可。完成之后可以将根文件系统按照我们的预期缩小 3G:

```
sh-3.2# resize2fs /dev/VolGroup00/LogVol00 5120M
resize2fs 1.39 (29-May-2006)
Resizing the filesystem on /dev/VolGroup00/LogVol00 to 1310720 (4k) blocks.
The filesystem on /dev/VolGroup00/LogVol00 is now 1310720 blocks long.

sh-3.2# fsck /dev/VolGroup00/LogVol00
fsck 1.39 (29-May-2006)
WARNING: couldn't open /etc/fstab: No such file or directory
e2fsck 1.39 (29-May-2006)
/dev/VolGroup00/LogVol00: clean, 145464/1310720 files, 874191/1310720 blocks
sh-3.2# _
```

完成之后强烈建议大家执行 fsck，一方面保证该文件系统没有受损，另外一方面也是最关键的是我们要获得这个文件系统目前的 size。根据提示，进行了 resize2fsz 的文件系统有 1310720 个块，每个块大小是 4k，那么整个文件系统大小即为 5120MB (1310720 x 4 / 1024)。

这个时候关键的问题来了，文件系统缩了，文件系统所在的逻辑卷到底应该收缩多少?! 这实际上是整个缩骨大法过程中最富有技术含量的问题了。前辈们告诉我等要多借助可能的系统工具获得目前 LVM 的信息，所以开另外一个终端执行 lvdisplay 以及 vgdisplay:

```
sh-3.2# lvm.static lvdisplay | more
--- Logical volume ---
LV Name                /dev/VolGroup00/LogVol00
VG Name                VolGroup00
LV UUID                wZQT2z-jISU-vOHM-XUw1-MDyE-ehL9-0vbQCW
LV Write Access        read/write
LV Status              available
# open                 0
LV Size                7.38 GB
Current LE             236
Segments              1
Allocation             inherit
Read ahead sectors    auto
 - currently set to    256
Block device           253:0

--- Logical volume ---
LV Name                /dev/VolGroup00/LogVol01
VG Name                VolGroup00
LV UUID                m2GZFP-91ot-wrWK-EK4Q-Prcn-lm1p-PLpmhI
LV Write Access        read/write
LV Status              available
# open                 0
LV Size                512.00 MB
--More-- _
```

```

    LV Size                512.00 MB
--More--
sh-3.2# lvm.static vdisplay
--- Volume group ---
UG Name                   VolGroup00
System ID
Format                    lvm2
Metadata Areas            1
Metadata Sequence No     3
UG Access                 read/write
UG Status                 resizable
MAX LV                    0
Cur LV                   2
Open LV                   0
Max PU                    0
Cur PU                   1
Act PU                    1
UG Size                   7.88 GB
PE Size                   32.00 MB
Total PE                  252
Alloc PE / Size           252 / 7.88 GB
Free PE / Size            0 / 0
UG UUID                   aj5pcQ-hnF6-k2aU-d4g5-CpUT-12Uw-CwNyhK

sh-3.2# _

```

从上面的两个截图显示，我的整个卷组是 252 个 PE，每个 PE 是 32MB，而在同一个卷组中所有 PE 和 LV 的大小默认是一致的，而且在同一个卷组中绝对不可能出现两种 size 的 PE 或者 LE。这样的话，我们即可以从 lvmstatic 命令显示的结果计算出目前根分区所在的 LVM 有多少个 4k 为单位的 block。

方法是 $236 \times 32 \times 1024 = 1933312$ ，即 1933312 个 4k 为单位的块，那么和已经收缩过的文件系统相减即可求出块设备和文件系统大小的差异： $(1933312 - 1310720) \times 4 / 1024 = 2432\text{MB}$ 。这就意味着，最多这个 LVM 可以缩 2432MB！

富于冒险精神的哥们当然可以缩更多，这是上帝赋予尔等的权力，之后的结果正好也可以让俺这种村里来的也张长见识。但鄙人胆小，所以照这个结果来缩！

```

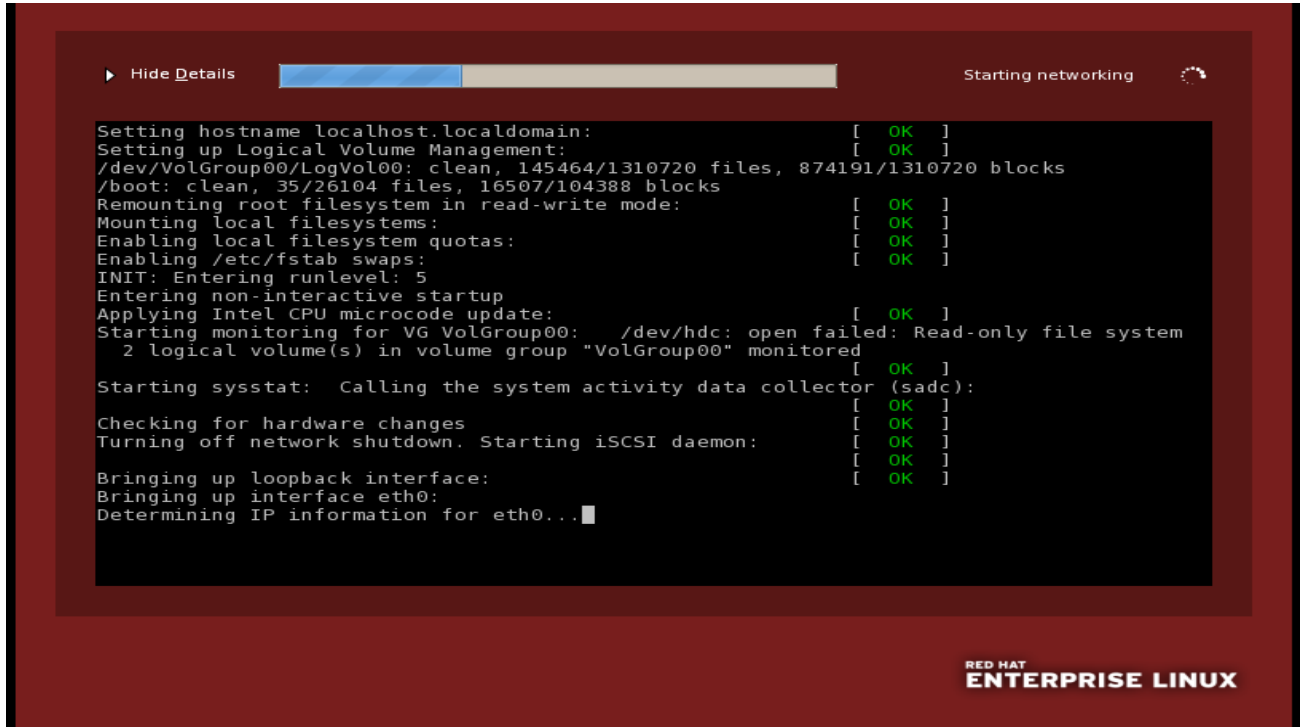
sh-3.2# lvm.static lvreduce -L -2432M /dev/VolGroup00/LogVol00
/dev/hdc: open failed: Read-only file system
WARNING: Reducing active logical volume to 5.00 GB
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce LogVol00? [y/n]: y
Reducing logical volume LogVol00 to 5.00 GB
Logical volume LogVol00 successfully resized
sh-3.2# resize2fs /dev/VolGroup00/LogVol00
resize2fs 1.39 (29-May-2006)
The filesystem is already 1310720 blocks long. Nothing to do!

sh-3.2# fsck /dev/VolGroup00/LogVol00
fsck 1.39 (29-May-2006)
WARNING: couldn't open /etc/fstab: No such file or directory
e2fsck 1.39 (29-May-2006)
/dev/VolGroup00/LogVol00: clean, 145464/1310720 files, 874191/1310720 blocks
sh-3.2# mkdir /temproot
sh-3.2# mount /dev/VolGroup00/LogVol00 /temproot/
sh-3.2# df -TH | grep /temproot
sh-3.2# df -TH | grep /temproot
          ext2      5.3G   3.5G   1.6G   69% /temproot
sh-3.2# _

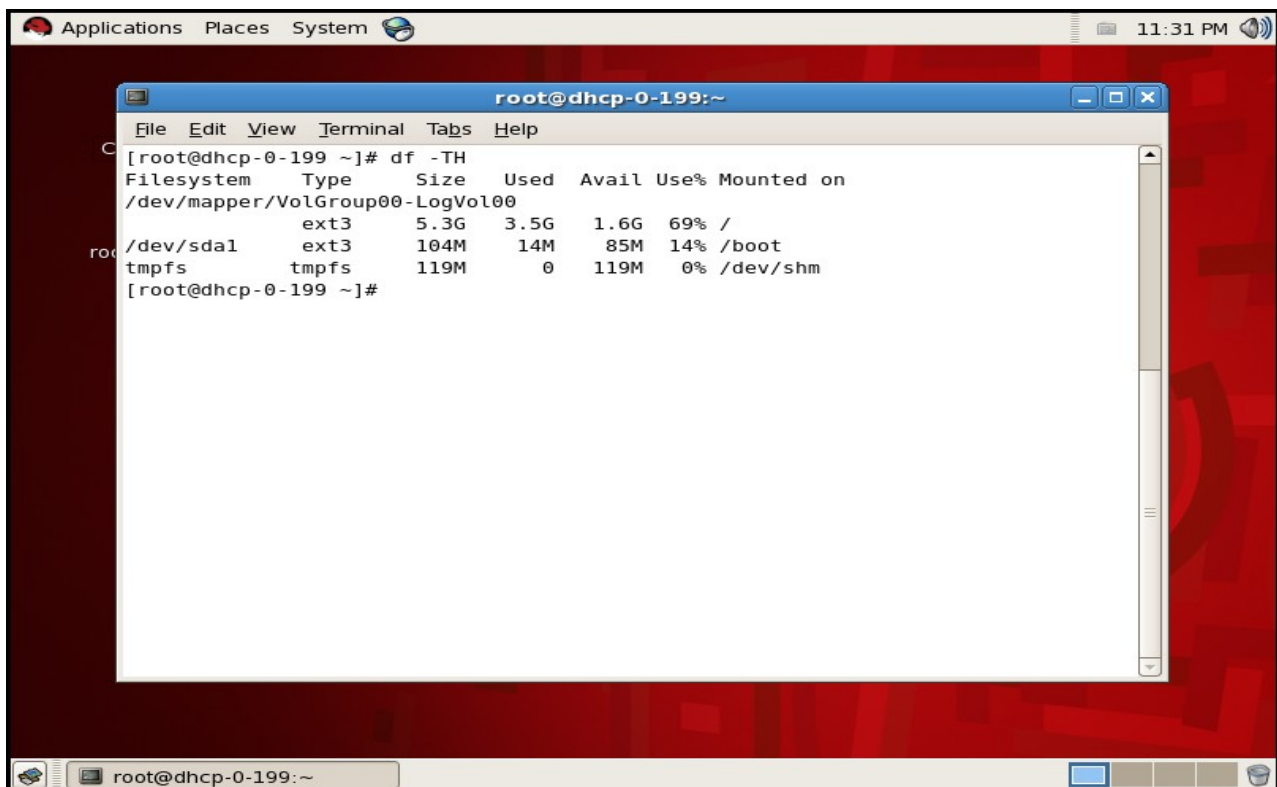
```

如图所示，执行 lvreduce 之后，再去 resize2fs，结果正好是吻合文件系统的大小 1310720 个 4k 块保险起见，跑一次 fsck 确保根文件系统没有问题，并挂载测试。结果一样能成功，根文件系统所在的 LV 实际上收缩了 2 个多 G。估计再多的话，根分区在 fsck 的时候即会有一堆的 error。

那么我们即可重启系统：



貌似一切正常，进入系统检查，真的一切正常啦！应该说操作成功了！



从上面的操作来看，本来文件系统有 8G，我在将其缩小到 5G 的时候，其所在的 LV 只能缩小 2 个多 G 才能保证文件系统安全。这也很容易解释，因为 Ext3 文件系统的布局处于优化性能的考虑是按照块组划分，每个块组中的块数量和结构基本上是一致的，那么前前后后可能会有一些不是我们通过常规计算所能获得的数值，所以实际上我们看到的 df 之后的结果，并不会是这个文件系统完全在物理上的 size。

不管怎么说，这次缩骨大法是成功了。有些人可能会问，如果我的根分区不在 LVM 上，而是在 sdx 上的某个分区，那么应该怎么整？！其实我可以负责任地告诉各位，操作的思想一样，而且因为没有了 LVM 在 rescue 环境中加载的问题，步骤相对其实会更加简单。有兴趣的哥们可以在自己的环境中测试一下，总不能什么玩意都指望在下。不过当然我从来说话给自己留后路，如果哪个虔诚的哥们愿意找我喝个茶，吃个包，沟通一下感情的同时探讨一下 LVM 什么的那就另说，如果是 MM 或者 PPMM 那就更另说。

但最后还是不忘唐僧一下：

* 系统部署之前考虑清楚自己今后的需求，不要顾前不顾后，顾头不顾尾。

* 没有迫不得已的需求，不要考虑缩根；如果要考虑缩根，不要考虑在线缩根；如果要在线缩根，不要……

算了，当我没说！

* 缩根的时候没敲一个回车之前都必须经过慎重和精确的计算，缓冲度尽量放大些，反正以后还可以执行 resize2fs，数学不好的哥们要慎重……

* 最后，如果因为各种各样的原因缩出了问题，那么我这唐僧就算成了佛也没法帮你，因此有条件务必做好备份先。善哉善哉！

实现 Linux 软件 RAID 的关键技术讨论

Chinaunix 网友：解宝琦/曹华

内容提要：

Linux 软件 RAID 技术是 Linux 操作系统提供的能够在软件层实现 RAID1、RAID5、RAID6、RAID10 等 RAID 磁盘阵列的有效技术手段，通过该技术我们可以建立起一个大容量的、具有很强数据冗余能力以及很高数据安全性的数据存储应用设备，可以通过该存储应用设备在网络中建立起初级的储存区域网(Storage Area Network, 简称 SAN)，本文通过在 Linux 中使用 6 块廉价 IDE 硬盘组建一个 RAID1+0 设备，配合 Linux 中的 VSFTP 服务器为客户端提供了安全、稳定、高效数据存储方案，实现 Linux 软件 RAID 的关键技术包括双网卡负载均衡实现网络连接的高可用性及数据流量的大吞吐量，RAID10 设备的有效创建方法和管理，RAID 设备 Shell 统一服务启动和终止脚本的创建。

1 引言

该应用方案通过使用一台普通的 P4 级的计算机以及 6 块移动硬盘，通过 LINUX 提供的软件 RAID 方法实现了一个大容量的、具有很强数据冗余能力以及很高数据安全性的数据存储案例，该应用案例在建设完成后，目前已经平稳使用两年，对入门级的数据存储具有很强的现实意义。

2 案例关键技术具体实现方法

2.1 双网卡绑定实现负载均衡

采用双网卡绑定技术实现服务器网络连接的高可用性及吞吐量。

1) 效果：网卡 eth0 和 eth1 虚拟成设备 bond0，并且向外公布一个虚拟 IP 地址（拓扑图如图 1），并提供两倍的网络带宽。其中一条链路或网卡出现故障时，系统应用不受故障影响。

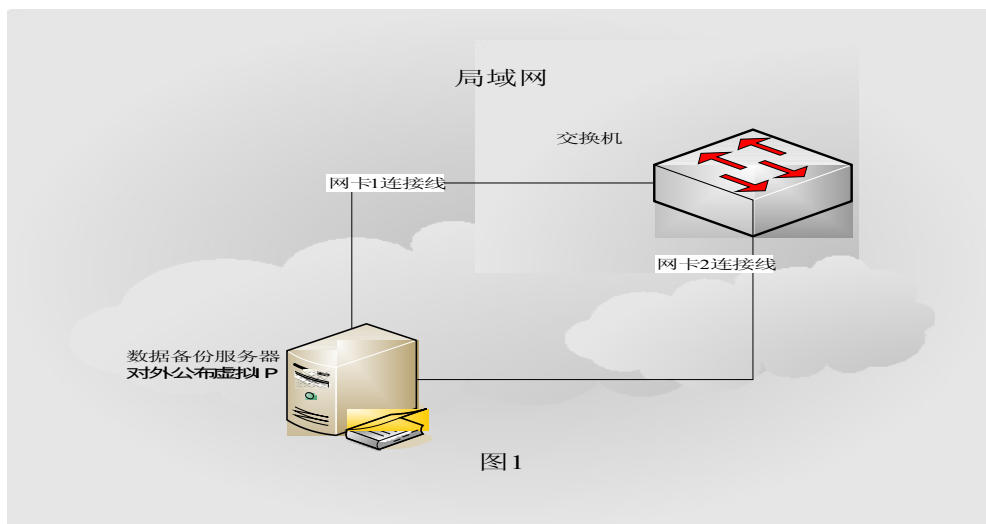
2) 配置方法：使用 VI 编辑器将标有//的内容添加到/etc/init.d/network 中的相应位置。（//不用输入，修改完后需要重启服务器操作系统）mode 为网卡工作方式，“1”为互援式工作方式（热备份工作方式，当前工作链路或网卡出现故障时，备份网卡会自动接管网络链路），“0”为集群式工作方式（同时在线提供两倍带宽，一条链路或网卡故障时，网络带宽为工作网卡带宽，不需切换）。示例如下：

```
//modprobe bonding mode=0 miimon=100 primary=eth0
//ifconfig bond0 inet 10.109.1.59 netmask 255.255.255.0
//route add default gw 10.109.1.1
//ifenslave bond0 eth0
//ifenslave bond0 eth1
//ifconfig bond0 up
;;
stop)
```

echo Shutting down network interfaces:

echo Shutting down network interfaces:

```
//ifenslave -d bond0 eth0
//ifenslave -d bond0 eth1
//ifconfig bond0 down
```



a) Linux 创建软 Raid 的配置过程 (RAID1+0)

在 Linux 中使用 6 块廉价 IDE 硬盘盒组建一个 RAID1+0 设备，从而实现数据的良好安全性。操

作系统启动正常后，分别依次开启 6 块廉价 IDE 硬盘盒电源开关，操作系统会依次加载这些设备，在系统提示符下输入 `sfdisk -s` 命令，系统返回如下信息：

```
/dev/sda: 78156288
/dev/sdb: 78150744
/dev/sdc: 78150744
/dev/sdd: 78150744
/dev/sde: 78150744
/dev/sdf: 78150744
/dev/sdg: 78150744
total: 547060752 blocks
```

其中 `/dev/sdb` 到 `/dev/sdg` 是系统为 6 块 IDE 硬盘创建的设备名称。我们将通过这些设备名称来使用该 6 块 IDE 硬盘创建 RAID1+0 设备。

第一步：创建新磁盘分区：

用 root 用户执行命令 `# fdisk /dev/hdb`，完成对第二个 IDE 硬盘的格式化。

在 Command (m for help): 提示下输入 `n`，即创建新的磁盘分区。

在 Command action

`e` extended

`p` primary partition (1-4): 提示下输入 `p`，即创建主分区。

在 Partition number (1-4): 提示下输入 `1`，即分区号

在 First cylinder (1-522, default 1): 提示下按回车，即指定起始柱面（注意：括号中尾柱面数字取决于硬盘的具体型号）

在 Last cylinder or +size or +sizeM or +sizeK (1-522, default 522): 提示下按回车，即指定尾柱面。

在 Command (m for help): 提示下输入 `p`，即查看分区状态，如果看到 `/dev/sdb1` 等信息则说明分区成功。

在 Command (m for help): 提示下输入 `t`，即转换分区类型

在 Hex code (type L list codes): 提示下输入 `L`，即显示分区编码列表

在 Hex code (type L list codes): 提示下输入 `fd`，即 LinuxRaid 分区编码

在 Command (m for help): 提示下输入 `w`，即保存退出

重复以上步骤，在其余五块硬盘上创建 LinuxRaid 分区。完成后需要重新启动 Linux 主机，以使设置生效。

第二步：创建 RAID 设备

执行创建 RAID1 设备命令

```
[root@localhost /]# mdadm -Cv /dev/md0 -l1 -n2 -c128 /dev/sd[b,c]1
[root@localhost /]# mdadm -Cv /dev/md1 -l1 -n2 -c128 /dev/sd[d,e]1
[root@localhost /]# mdadm -Cv /dev/md2 -l1 -n2 -c128 /dev/sd[f,g]1
```

命令中各参数的作用如下：

“-C”：创建一个新的阵列；“/dev/md0”：表示阵列设备名称；“-l1”表示设置阵列模式，可以选择 0、1、4、5、6，它们分别对应于 RAID0、RAID1、RAID4、RAID5、RAID6，这里设为 RAID1 模式；“-n2”指设置阵列中活动设备的数目，该数目加上备用设备的数目应等于阵列中的总设备数；“-c128”指设置块的大小为 128KB，缺省为 64KB；“/dev/sd[b,c]1”指当前阵列中包含的所有设备标识符，也可以分开来写，中间用空格分开。

当创建完成后，查看 /proc/mdstat 文件：

```
Personalities : [raid1] [raid0]
md2 : active raid1 sdg1[1] sdf1[0]
      78148096 blocks [2/2] [UU]
md1 : active raid1 sde1[1] sdd1[0]
      78148096 blocks [2/2] [UU]
md0 : active raid1 sdc1[1] sdb1[0]
      78148096 blocks [2/2] [UU]
unused devices: <none>
```

执行创建 RAID0 设备命令

```
[root@localhost etc]# mdadm -Cv /dev/md3 -l0 -n3 -c128 /dev/md0 /dev/md1 /dev/md2
```

查看 /proc/mdstat 文件出现如下信息：

```
Personalities : [raid1] [raid0]
md3 : active raid0 md2[2] md1[1] md0[0]
      234443904 blocks 128k chunks
md2 : active raid1 sdg1[1] sdf1[0]
      78148096 blocks [2/2] [UU]
md1 : active raid1 sde1[1] sdd1[0]
      78148096 blocks [2/2] [UU]
md0 : active raid1 sdc1[1] sdb1[0]
      78148096 blocks [2/2] [UU]
unused devices: <none>
```

至此，RAID10 盘阵创建完成。

第三步：配置 RAID 设备

接下来我们为 mdadm 生成配置文件，mdadm 的缺省配置文件为 /etc/mdadm.conf，它主要是为了方便阵列的日常管理而设置的，对于阵列而言不是必须的，但是为了减少日后管理中不必要的麻烦，还是应该坚持把这一步做完。使用 vi 命令，按照规定的格式编辑修改 /etc/mdadm.conf 文件 文件内容如下：

```
ARRAY /dev/md0 level=raid1 num-devices=2
UUID=c4f17506:687b1bd2:9964e709:c0d4630d devices=/dev/sdb1,/dev/sdc1

ARRAY /dev/md1 level=raid1 num-devices=2
UUID=ae52d48a:e65c88c2:0bb4c763:50499806 devices=/dev/sdd1,/dev/sde1

ARRAY /dev/md2 level=raid1 num-devices=2 UUID=4027a9e7:a8fab70b:272f6793:c8a72f12
devices=/dev/sdf1,/dev/sdg1
```

```
ARRAY /dev/md3 level=raid0 num-devices=3
UUID=1cdfe4da:d88fe61e:54d2ed07:7aeb9c96 devices=/dev/md0,/dev/md1,/dev/md2
```

//每行的 devices 后的内容为添加内容，设备名以盘阵的设备名为准。可以使用命令 mdadm -Ds 来获得操作系统给出的文件内容信息。

重启服务器后，按照顺序依次打开硬盘盒，执行如下命令：

```
[root@localhost etc]# mdadm -As
```

出现如下信息：

```
mdadm: /dev/md0 has been started with 2 drives.
mdadm: /dev/md1 has been started with 2 drives.
mdadm: /dev/md2 has been started with 2 drives.
mdadm: /dev/md3 has been started with 3 drives.
```

说明系统已经挂载完成创建好的 RAID10 阵列。

第四步：生成文件系统

创建文件系统并挂接(mount)使用。RAID10 已经启动并处于运行状态，现在要做的就是在其上创建一个文件系统，这里使用 mkfs 命令，文件系统类型为 ext3。命令如下：

```
# mkfs.ext3 /dev/md3 //以 ext3 文件系统格式化 RAID10 设备
```

当新的文件系统生成之后，就可以将 /dev/md0 挂接到指定的目录了。命令如下：

```
# mount /dev/md3 /ftp //挂接/dev/md0 设备到/ftp 目录
```

2.3 编写统一服务启动和终止脚本

在 /etc/init.d/ 中分别建立 mdmountvsftpd_start 和 mdmountvsftpd_stop 文件。过程如下：

```
[root@localhost init.d]# vi mdmountvsftpd_stop （终止服务脚本）
```

Mdmountvsftpd_stop 内容如下：

```
#!/bin/bash
```

umount /ftp //卸载/ftp 挂载设备

mdadm --stop /dev/md3 //停止使用/dev/md3 设备

mdadm --stop /dev/md2 //停止使用/dev/md2 设备

mdadm --stop /dev/md1 //停止使用/dev/md1 设备

mdadm --stop /dev/md0 //停止使用/dev/md0 设备

修改 mdmountvsftpd_stop 文件权限

```
[root@localhost init.d]# chmod 700 mdmountvsftpd_stop
```

```
[root@localhost init.d]# vi mdmountvsftpd_start (启动服务脚本)
```

Mdmountvsftpd_start 内容如下:

```
#!/bin/bash
```

```
mdadm -As //加载/etc/mdadm.conf 文件中存在的 RAID10 设备
```

```
mount /dev/md3 /ftp //加载/dev/md3RAID10 设备到/ftp 目录
```

修改 mdmountvsftpd_start 文件权限

```
[root@localhost init.d]# chmod 700 mdmountvsftpd_start
```

3 结论与展望

在此我们使用 Linux 提供的双网卡绑定技术以及软件 RAID 技术, 使用普通的 PC 机和 IDE 硬盘盒组成了一个性能良好的存储设备, 我们可以将该 RAID10 设备服务器部署为一般的 WEB、FTP、文件共享应用案例的存储设备解决方案, 也可以将该 RAID10 设备服务器部署为 ISCSI 应用服务器在 HA、集群应用等需要共享盘阵的更高级应用部署中作为快速的、有效的、安全的、健壮存储设备来使用, 更可以将该设备作为廉价储存区域网(Storage Area Network, 简称 SAN) 设备进行部署。

注意: 系统开机顺序:

- 1、 启动操作系统
- 2、 操作系统正常启动后依次打开硬盘盒电源
- 3、 执行/etc/init.d/mdmount_start 命令启动 RAID10 阵列服务,挂载目录/ftp, 启动 VsftpdFTP 服务

关机顺序:

- 1、 执行/etc/init.d/mdmount_stop 命令终止 VsftpdFTP 服务,卸载目录 ftp, 终止 RAID10 阵列服务
- 2、 执行 shutdown -h now 命令关机
- 3、 操作系统正常关闭后依次关闭硬盘盒电源

问题: 硬盘盒的挂载必须在操作系统启动后按顺序打开电源, 这样才能保证盘序正确, 并顺利启动 RAID10 服务。不知道哪位能给出自动按照盘序挂载的方法 (即确保盘序每次都能按照固定的顺序和硬盘盒对应) 。

VMware 环境下 OpenBSD 4.5 以及 Gnome 2.24 的安装

ChinaUnix 网友: OraBSD

Part I OpenBSD 4.5 的安装

1. 什么是 OpenBSD?

OpenBSD 是一个自由的、基于 4.4BSD 的支持多种平台的类 UNIX 操作系统。OpenBSD 项目组致力于可移植性、标准化、正确性，前瞻的安全性和完整的加密技术。OpenBSD 支持 SVR4(Solaris), FreeBSD, Linux, BSD/OS, SunOS 和 HP-UX 下大多数程序的二进制仿真运行。

2. VMware 下安装 OpenBSD 的准备

工欲善其事，必先利其器。下面列出笔者这次实验的安装环境：

硬件环境：ASUS X50Mseries F5M 笔记本，

内存：2.5GB，

CPU:AMD Truion64 2.01GHz，

显存：256MB，

硬盘：80GB。

软件环境：Windows XP SP2，

VMware 6.5.1，

DAEMON Tools Lite 4.30.3（即虚拟光驱，可以不用，后面我会说明）。

OpenBSD 安装介质：install45.iso；

OpenBSD 官方网站下载的 ISO 镜像文件，

下载地址：<ftp://ftp.openbsd.org/pub/OpenBSD/4.5/>，在里面选择您需要的平台版本，本次安装使用的 AMD64 版本。所以进入 amd64,即

<ftp://ftp.openbsd.org/pub/OpenBSD/4.5/amd64/>，下载 [install45.iso](#) 到本地磁盘即可。

3. OpenBSD 的安装

VMware 的安装很简单，网上也很多相关教程，由于不是本文主题，所以就不做介绍了。下面进入主题。

鼠标左键单击 windows 桌面左下角“开始菜单”，将光标移至“程序”->“VMware”->“VMware Workstation”，鼠标左键单击打开 VMware 软件。如图 01 所示：

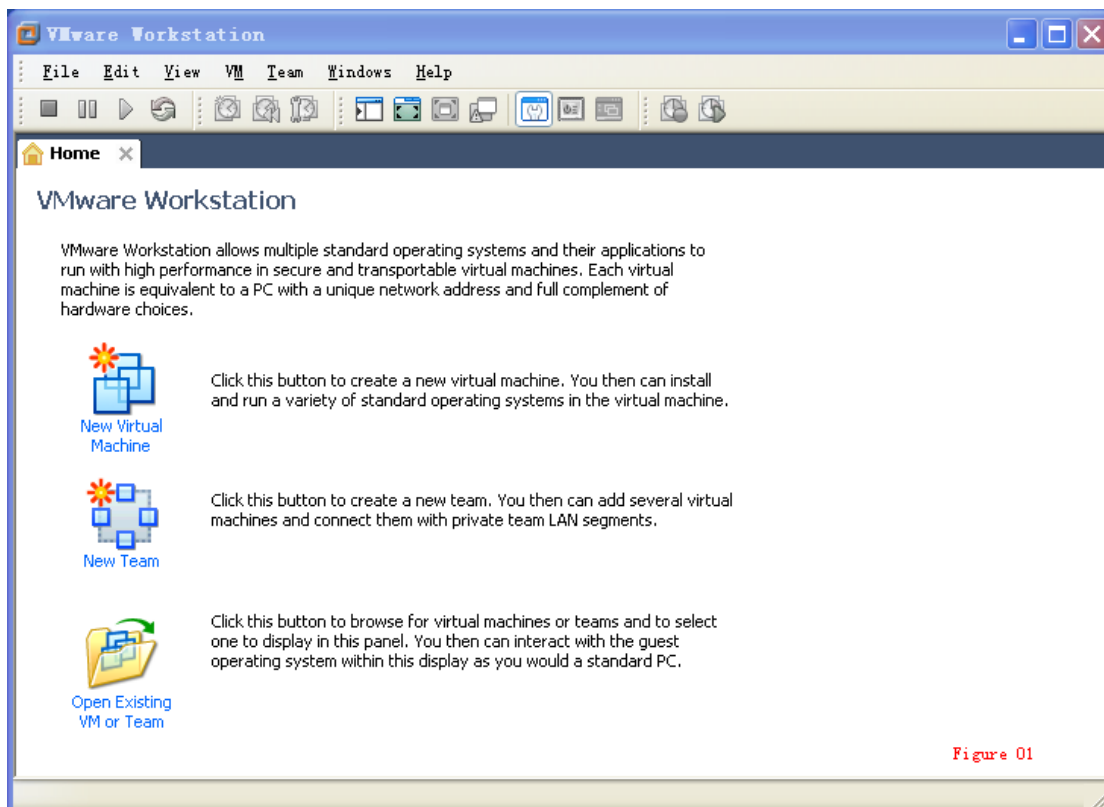


图 01

然后鼠标左键单击“New Virtual Machine”,即“新建虚拟机”。出现如图 02 所示界面:

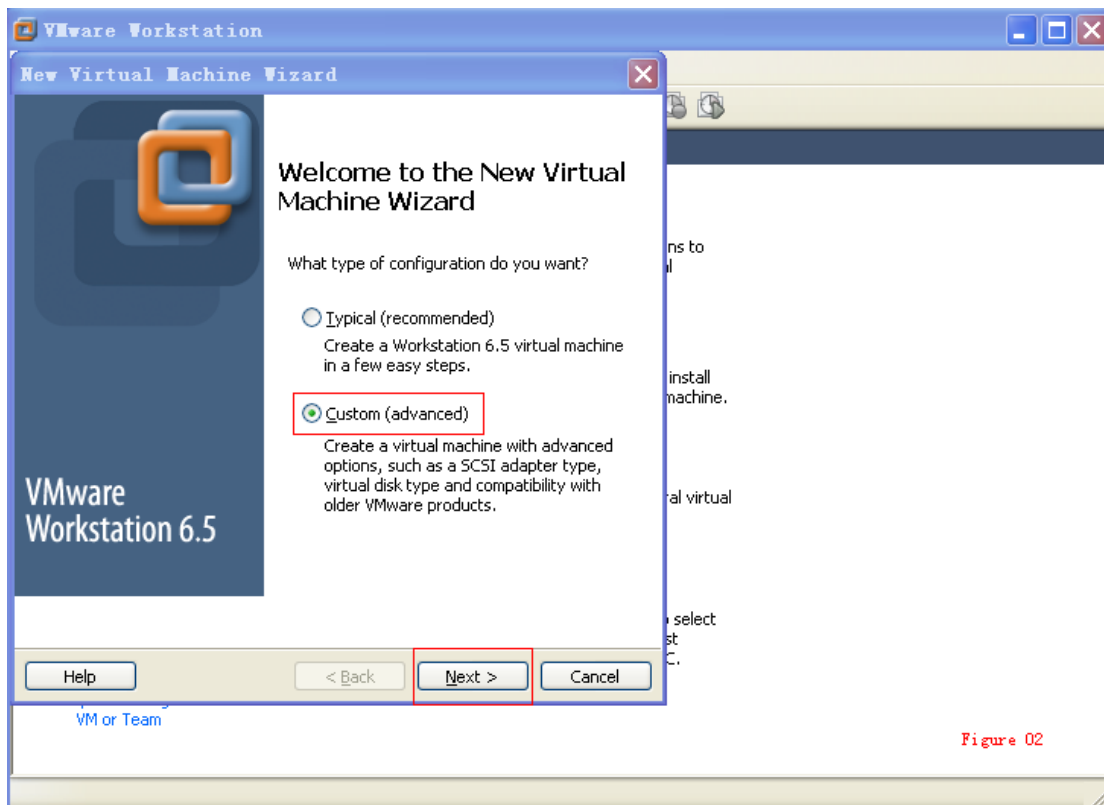


图 02

为了方便我们自己设置，我们选择“Custom(advanced)”，鼠标左键单击那个小圆点。然后鼠标左键单击“Next>”，即“下一步”。出现如图 03 所示界面：

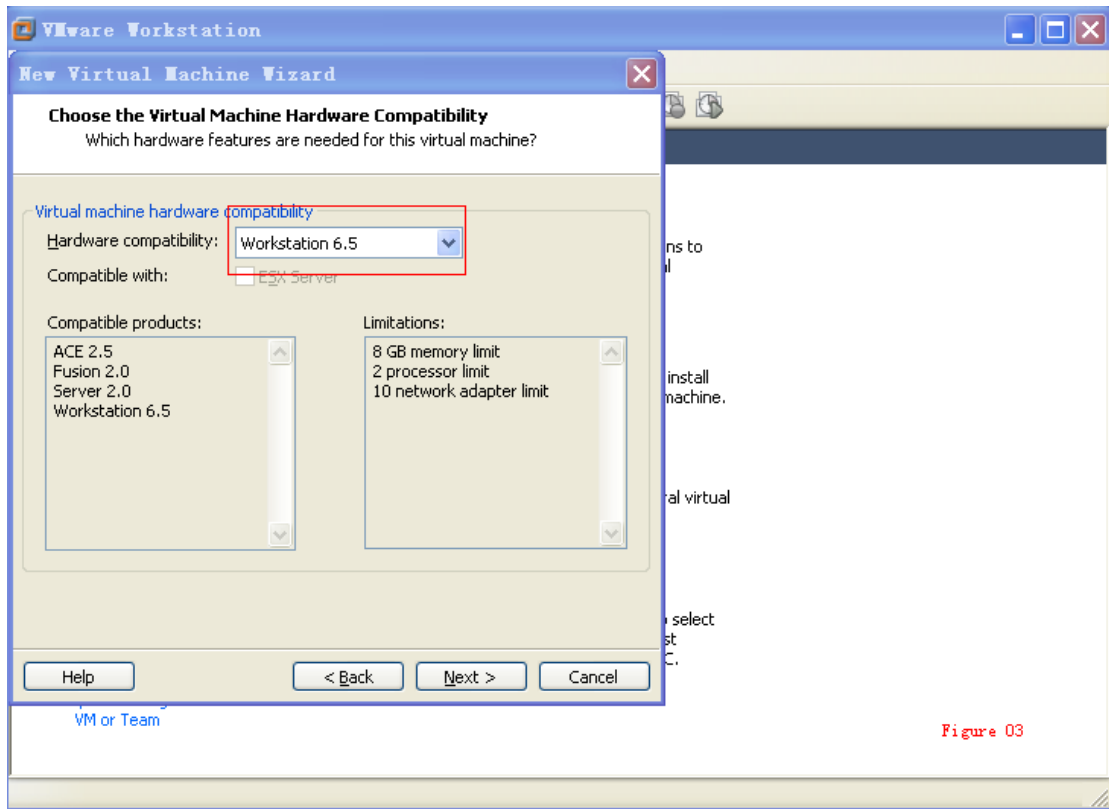


Figure 03

图 03

这里我们选择安装 6.0 版的，因为 6.0 版的虚拟机更节省资源。还有一个原因就是 OpenBSD 如果在 VMware 6.5 版的虚拟机下安装会出现一些未知设备（缺少相关虚拟设备驱动），虽然没什么影响，但是看着不爽。为了使得看起来完美，还是选择 6.0 的好。在这里多说两句，VMware 6.5 就是修补了一些 6.0 版本的重大漏洞，技术上并没有多大提升，只是增强了 3D 显示功能，意义不大，反而更加消耗内存，内存低于 2G 的朋友还是推荐选择虚拟一个 6.0 的。见图 04：

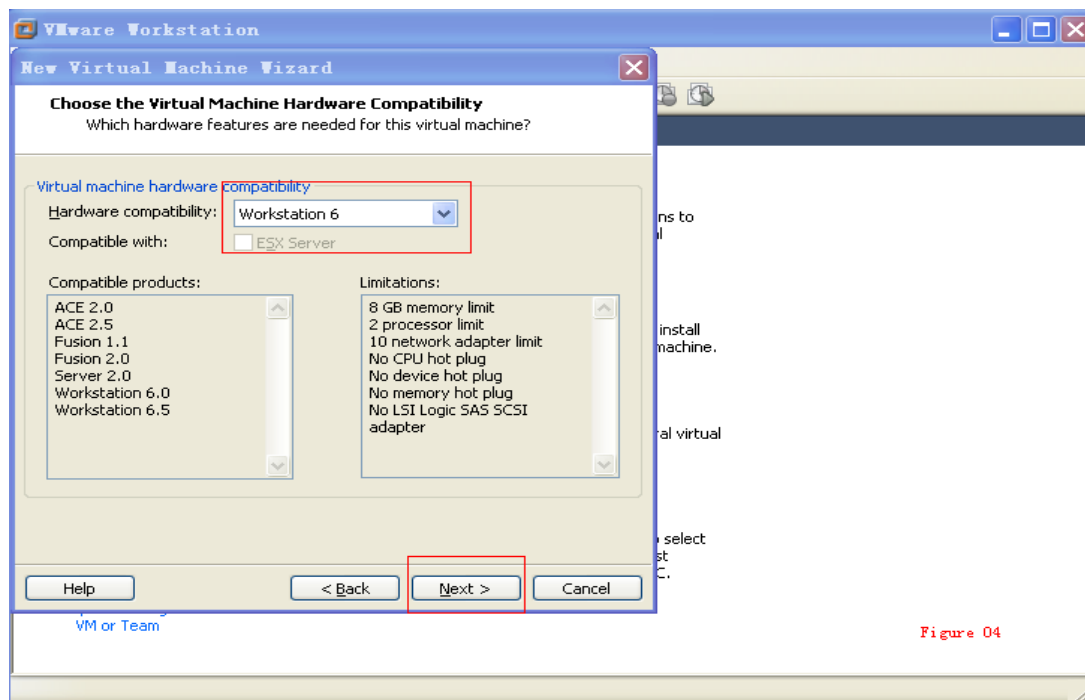


图 04

然后鼠标左键单击“Next>”,即“下一步”,见图 05:

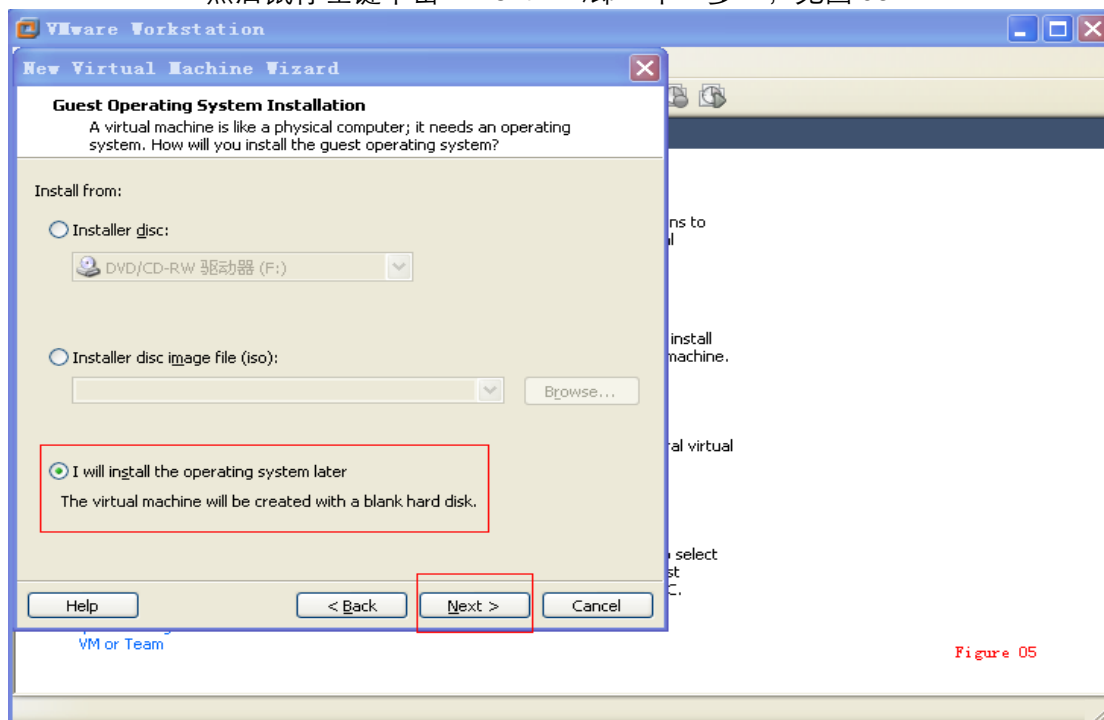


图 05

在这里我们看见“Install from”(安装来源)下面有三个选项: 1. “Install disc”选项就是选择光盘安装的, 购买了 OpenBSD 光盘的朋友可以选择这个选项(购买 OpenBSD 的安装光盘能够支持 OpenBSD 项目的继续发展, 也使得我们以后能够继续使用 OpenBSD 这个优秀的操作系统); 另外, 安装了虚拟光驱而使用 ISO 镜像文件的朋友也可以使用这个选项, 前提是您已经开启了虚拟光

驱软件，并且加载了 OpenBSD 的安装 ISO 文件，关于虚拟光驱的操作我在后面介绍。2. “Install disc image file(iso)”，如果选择此选项的话，就不要虚拟光驱了，具体操作详见后面。3. “I will install the operating system later”，即在创建了虚拟机后在安装，这里我们选择此选项。然后鼠标左键单击“Next>”，即“下一步”，见到图 06：

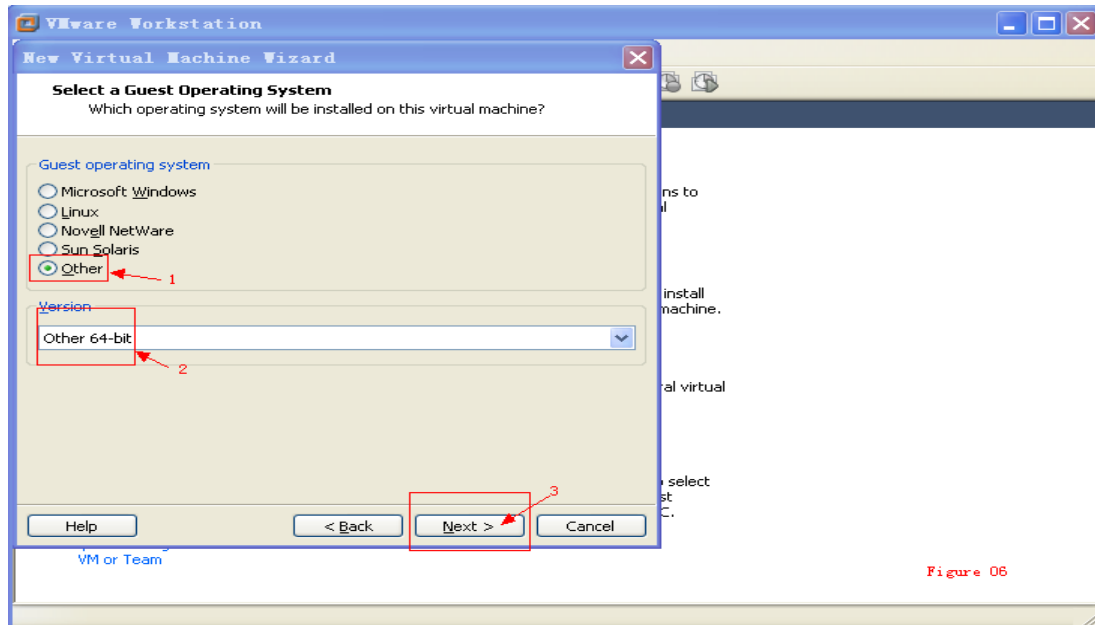


图 06

在图 06，我们鼠标左键单击“Guest Operating system”（客户操作系统）下面“other”（其他）左边的小圆圈，表示选中“other”（其他），然后再在“version”点鼠标左键单击那个倒三角的下拉按钮，选择“Other 64-bit”，然后鼠标左键单击“Next>”，见图 07 所示：

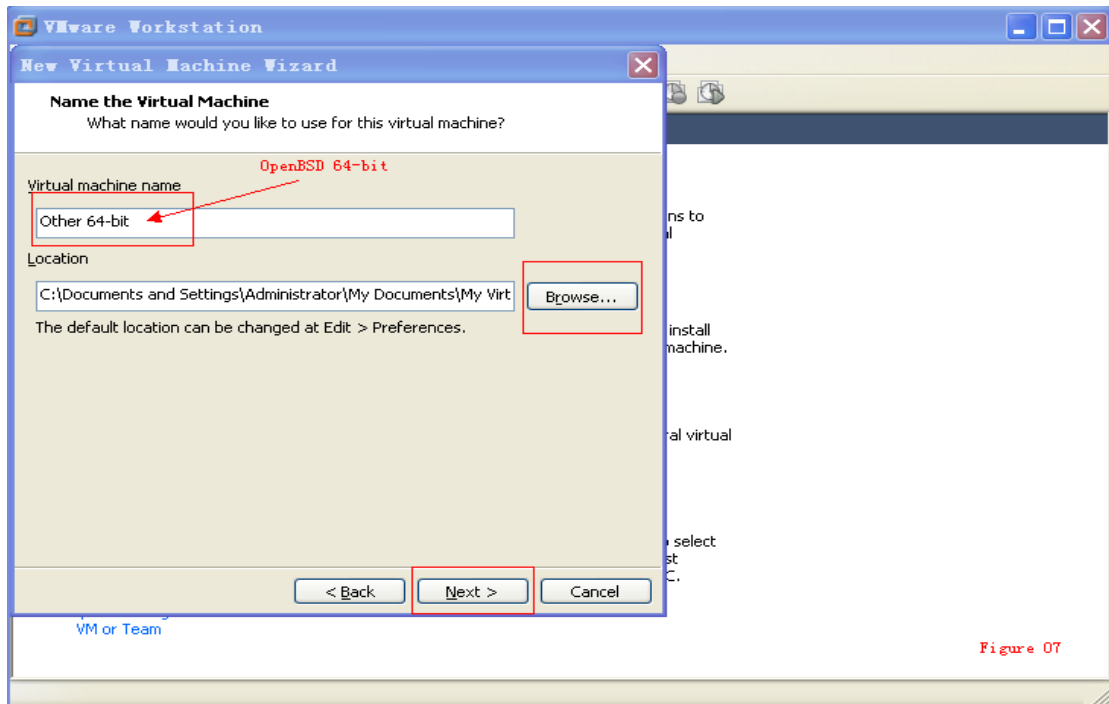


图 07

为了方便我们以后快速找到需要的虚拟机，在“Virtual machine name”（虚拟机名字）下面的方框里把名字改为“OpenBSD 64-bit”（OpenBSD 64 位）；然后在“Location”下面确定虚拟机创建在磁盘中的路径，即创建位置。可以手动输入具体位置，如果没有相应文件夹的话，它会自动创建，也可以点击右边的“Browse”（浏览）选择具体路径，参见图 08：

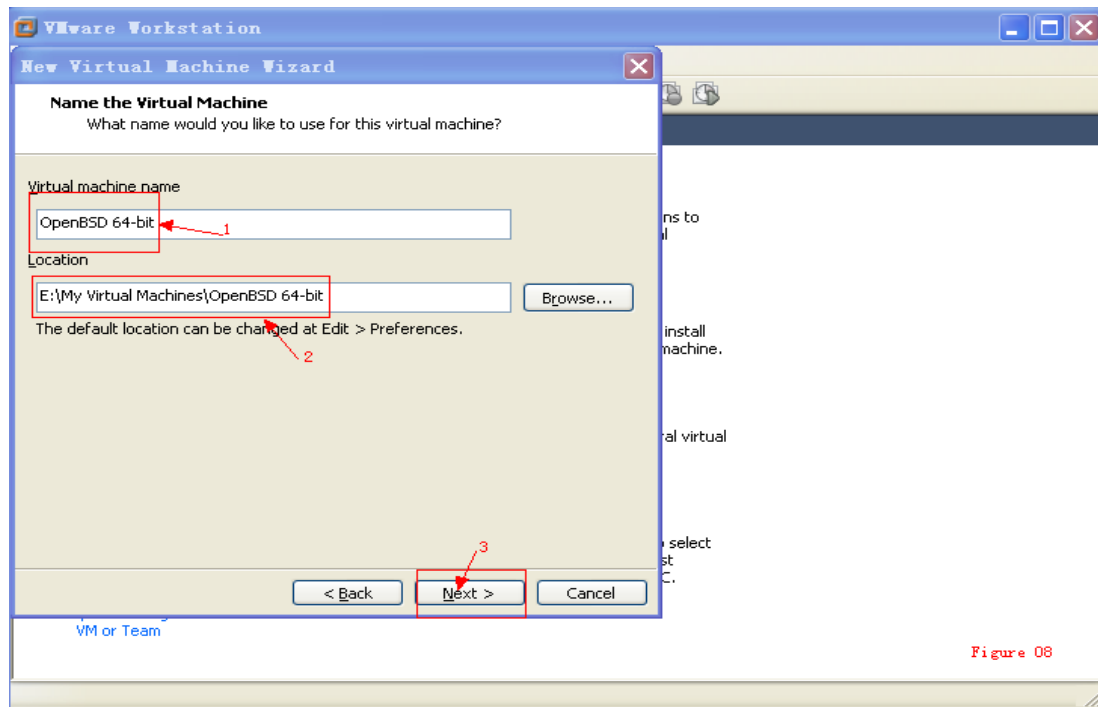


图 08

然后鼠标左键单击“Next>”（即“下一步”）。见图 09：

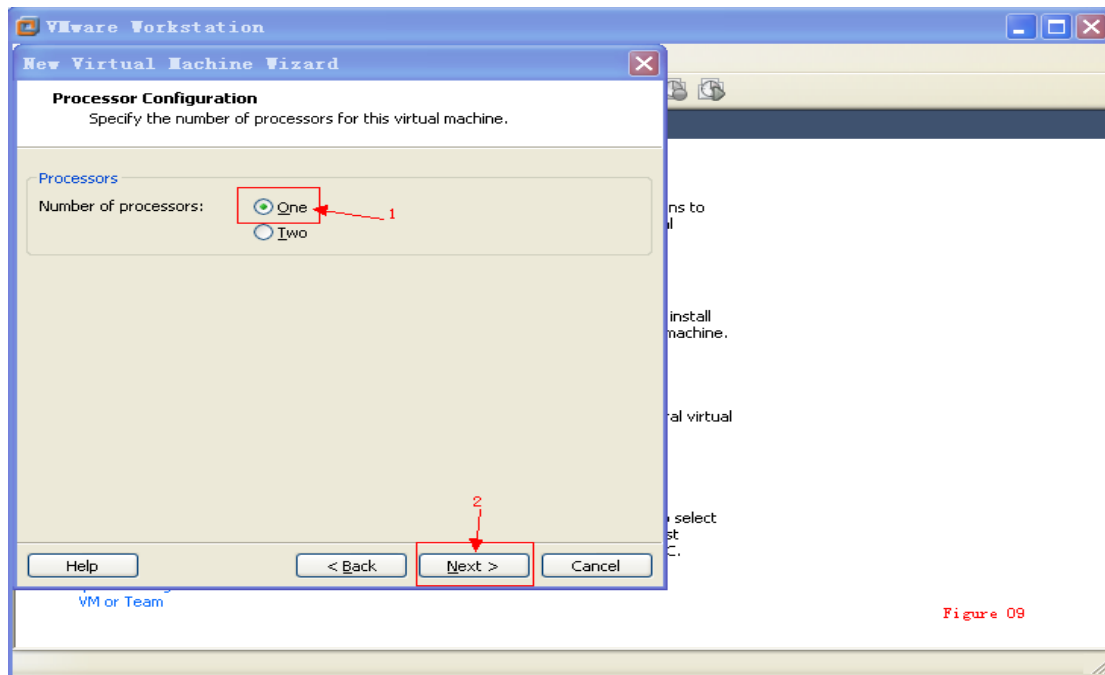


图 09

因为我的笔记本是单核的，所以在“Processor”下面的“Number of processor”里选择“One”，然后鼠标左键单击“Next>”（“下一步”）。出现如图 10 所示界面：

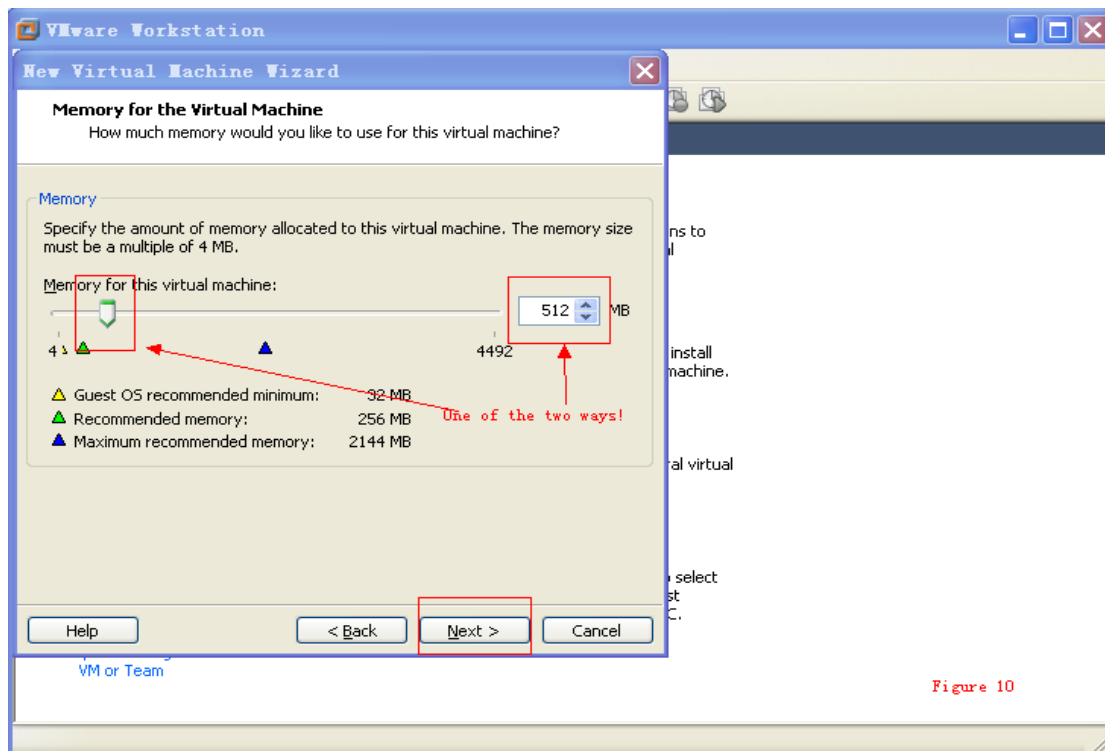


图 10

Memory（内存）大小可以自己设置，随时可以修改。在这里我选择了 512MB。内存大小选择 4MB 的整数倍。然后选择“Next>”（“下一步”）。参加图 11：

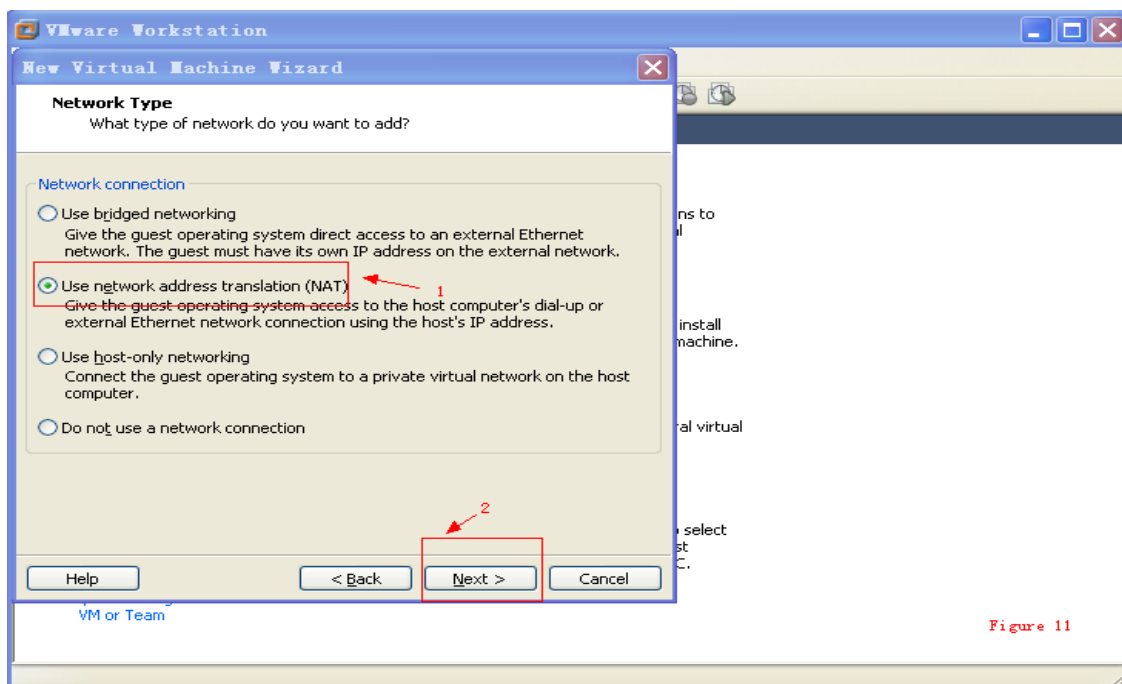


图 11

下面设置虚拟机的联网方式, 这个也可以在以后随时修改。在这里我们选择使用 NAT (默认) 方式。

★★★说明: VMware 联网的几种方式:

use bridged networking:这种方式使得虚拟机系统拥有 1 个和真机系统(是指与虚拟机系统同在一台电脑上的真实环境中的系统)不一样的 IP 地址,从而与网路上的任何电脑互相通信,默认使用虚拟网卡 VMnet0。

use network address translation(NAT):这种方式使得虚拟机系统拥有 1 个和真机系统一样的 IP 地址,实现虚拟机系统和真机系统互相通信.虚拟机系统可以访问网路上的其他电脑,而网路上的其他电脑不能访问虚拟机系统,默认使用虚拟网卡 VMnet8。

use host-only networking:这种方式只实现虚拟机系统和真机系统互相通信,不能与网路上的其他电脑通信,默认使用虚拟网卡 VMnet1。

※注: 为了节约时间, 以上说明来自互联网。其实 VMware 自带的帮助文档里也有详细的介绍。有兴趣的朋友可以自己参考。

下面接着进行, 选择 “Next>” (“下一步”),见图 12:

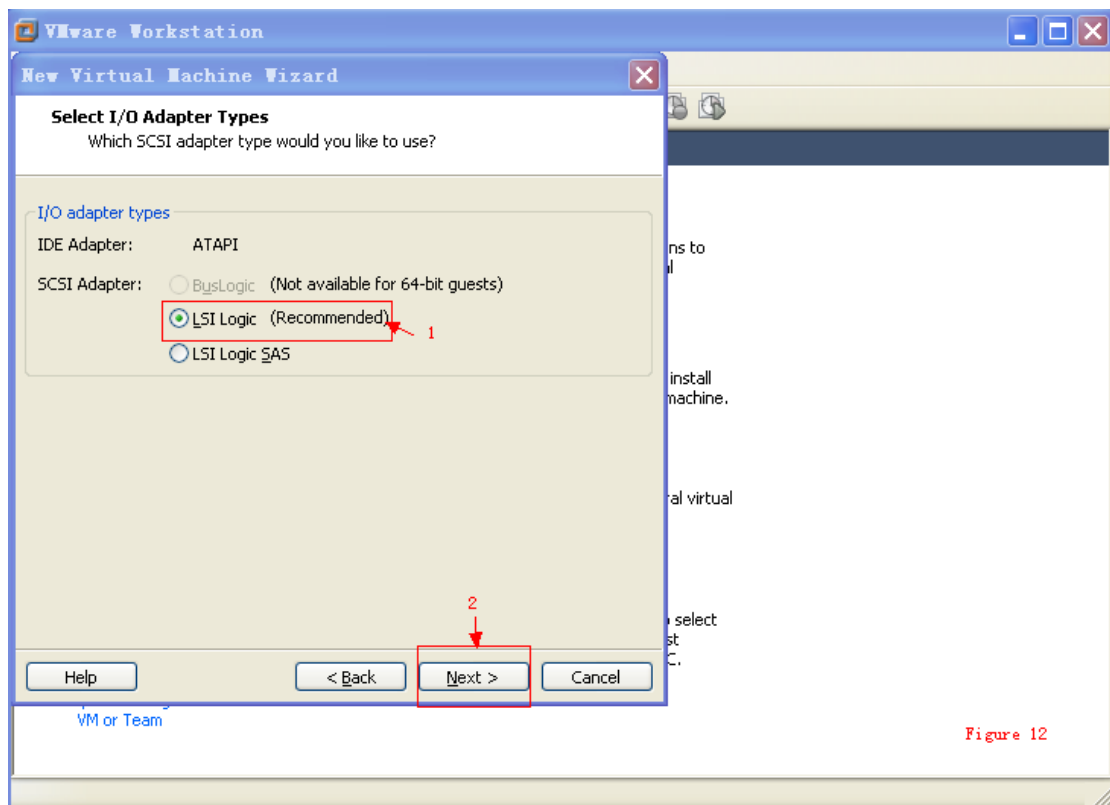


图 12

选择默认设置, 直接下一步, 见图 13:

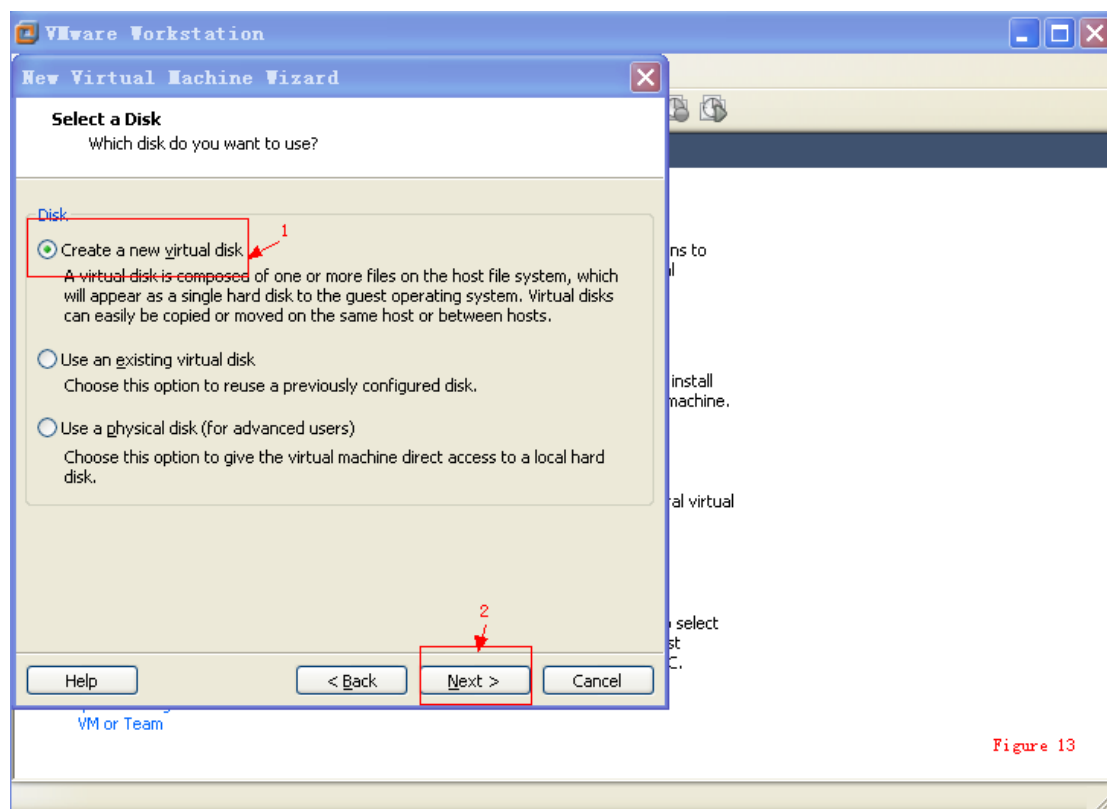


图 13

依然是下一步，见图 14：

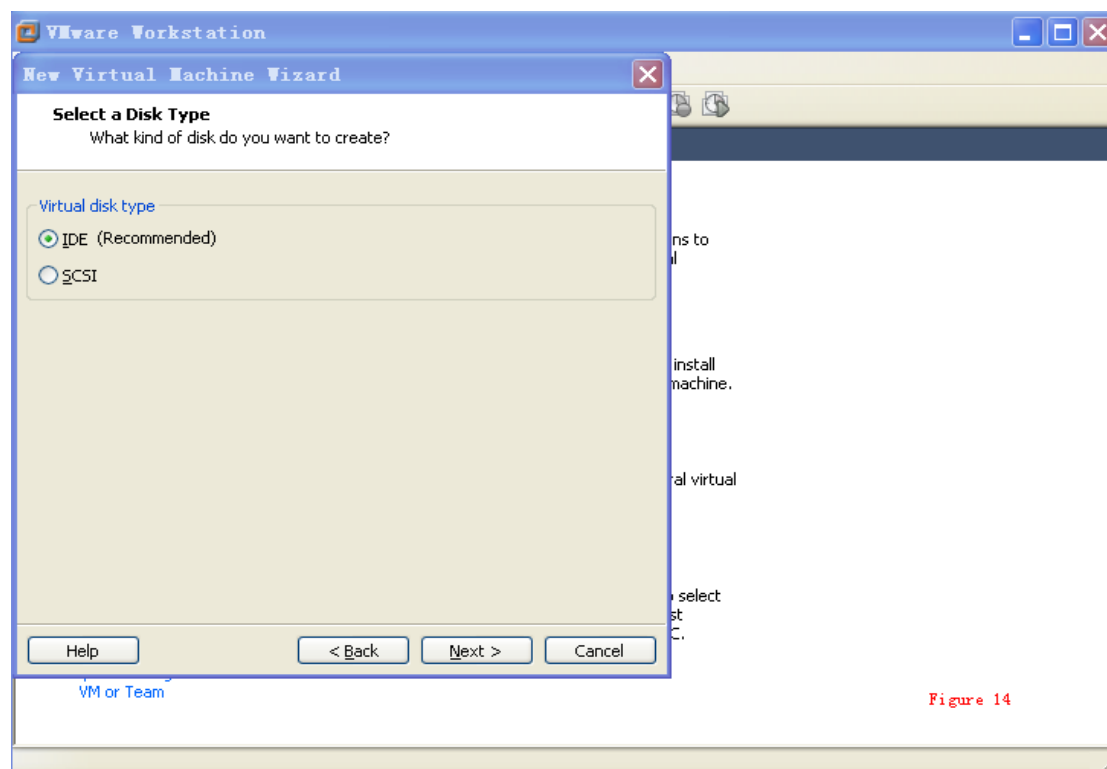


图 14

下一步，见图 15：

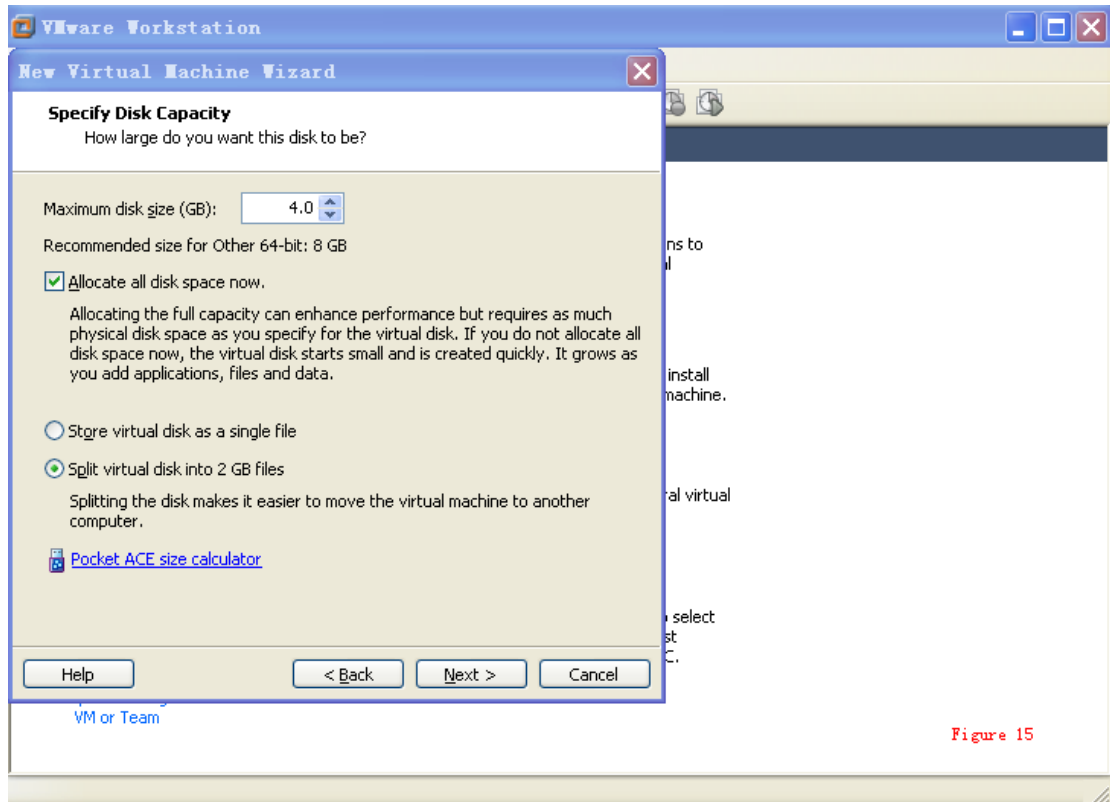


图 15.

※注意，为了创建虚拟磁盘节约时间我选择了 4GB 的虚拟磁盘大小，此处视您的具体情况而定，如果今后要在 OpenBSD 里安装更多应用软件或者 Gnome 之类的重量级桌面，至少准备 6GB 硬盘。为了提升性能，我们选择“Allocate all disk space now”，并且为了便于 Windows 作磁盘碎片整理（因为虚拟机创建的虚拟磁盘全是碎片文件，如果选择一个 8GB 大小或者更大的虚拟磁盘并且又不分割成单个 2GB 大小文件，而您的硬盘（虚拟机所在硬盘分区）不够大的话，到时候作磁盘碎片整理都不行，Windows 做磁盘整理时，被整理的盘分区至少需要该盘 15% 的剩余空间），也为了方便今后移动文件，将虚拟磁盘分割成个 2GB 大小，选择“Split virtual disk into 2 GB files”，然后进行“Next>”（“下一步”），见图 16：

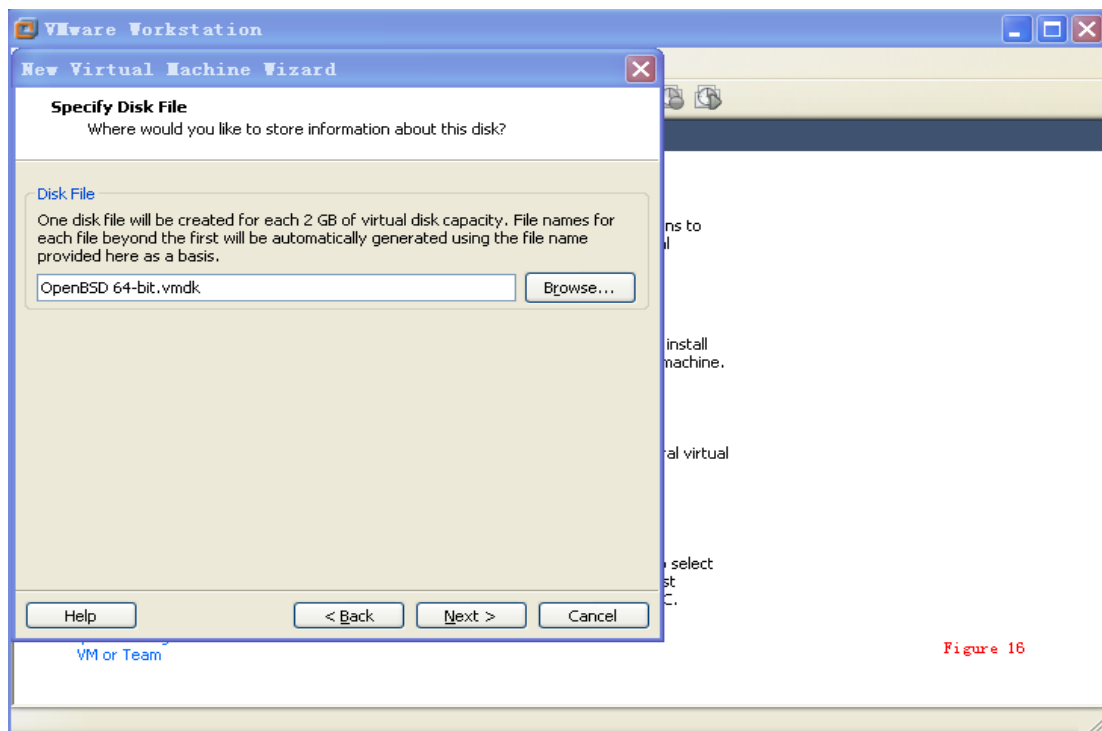


图 16

下一步，见图 17：

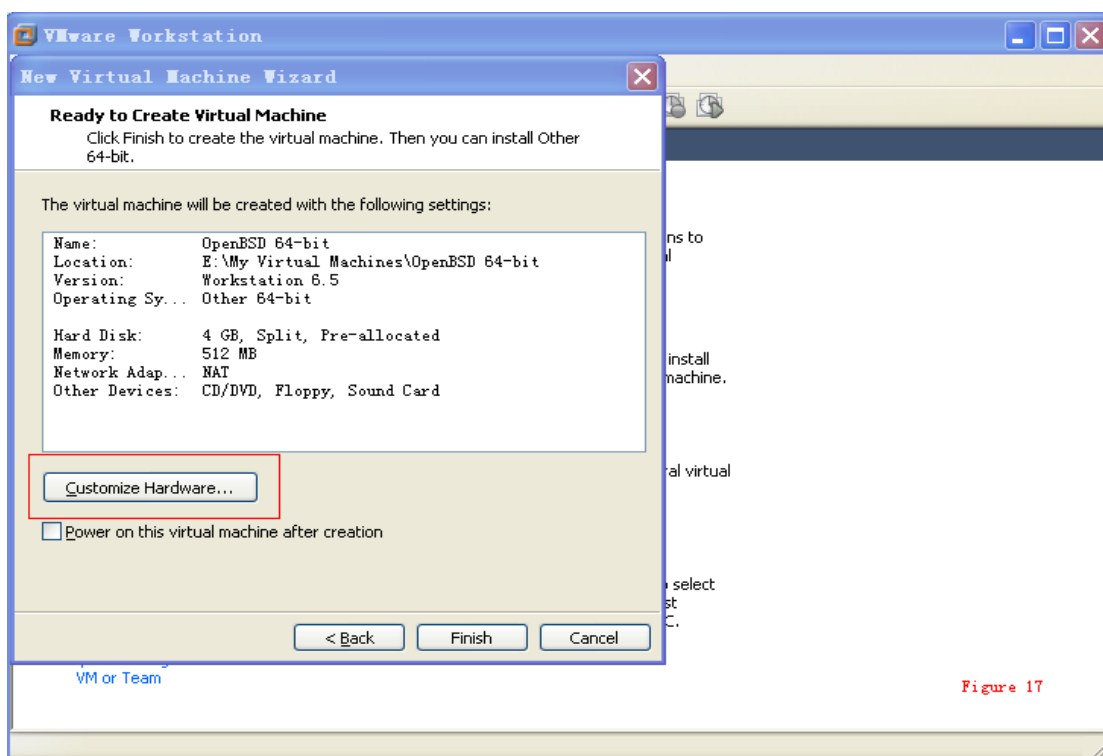


图 17

此处是选择的最后一步，如果没有需要改动的话，就鼠标左键单击“Finish”开始创建虚拟机了。

附：上面不是有个“Custom Hardware”（自定义硬件）么？选择它可以添加硬盘，并口，串口

之类的虚拟硬件设备。当然，也可以通过它删除一些不必要的设备，比如那个软驱“Floppy”就没有什么用处，呵呵。还有一个就是那个“Power on this virtual machine after creation”选项，没必要选，这样更自由^_^。

OK，基本设置搞定，开始创建虚拟机吧。看看创建过程，图 18：

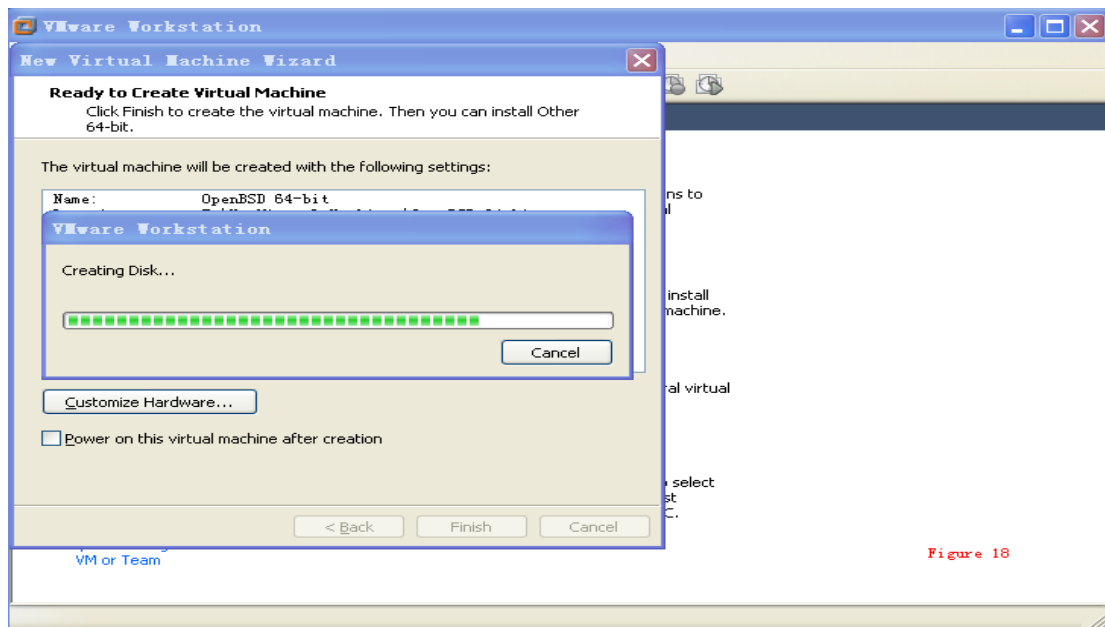


图 18

创建好后如图 19 所示：

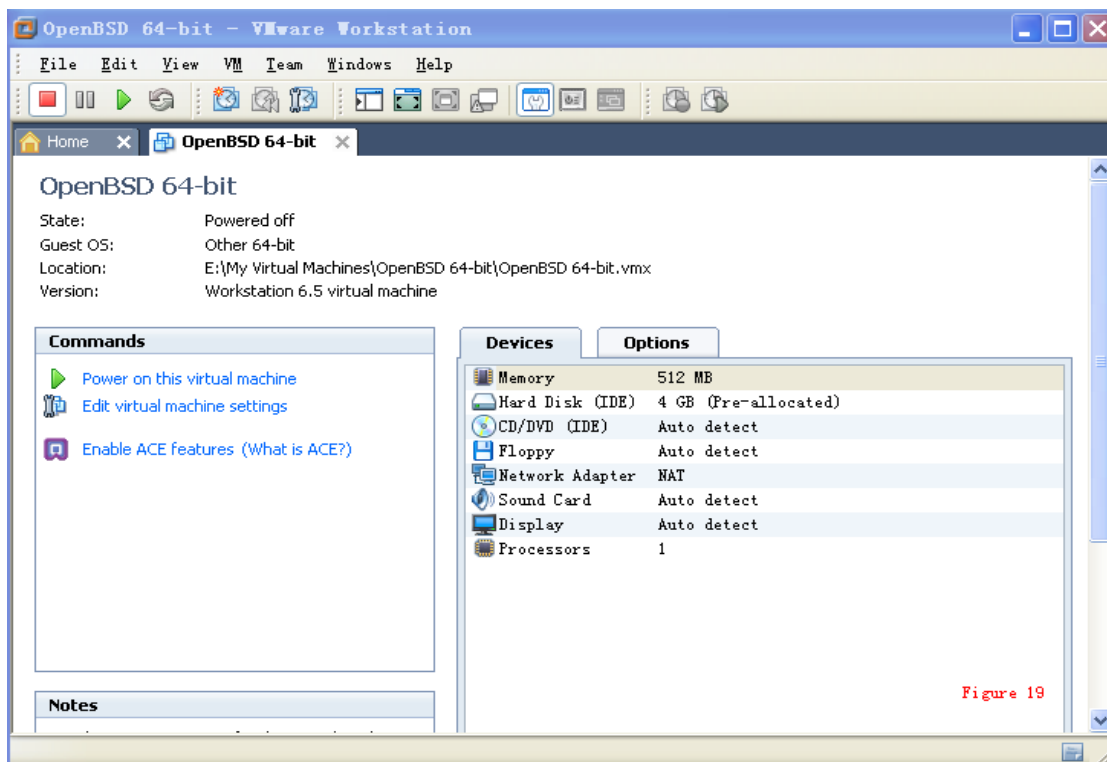


图 19

OK，下一步就是真正的安装过程了。

前面我们提到选择安装介质的问题。现在我作如下回答：

图 19 上面可以看到一个“Edit virtual machine settings”（编辑虚拟机设置项），鼠标左键单击此处。见图 20：

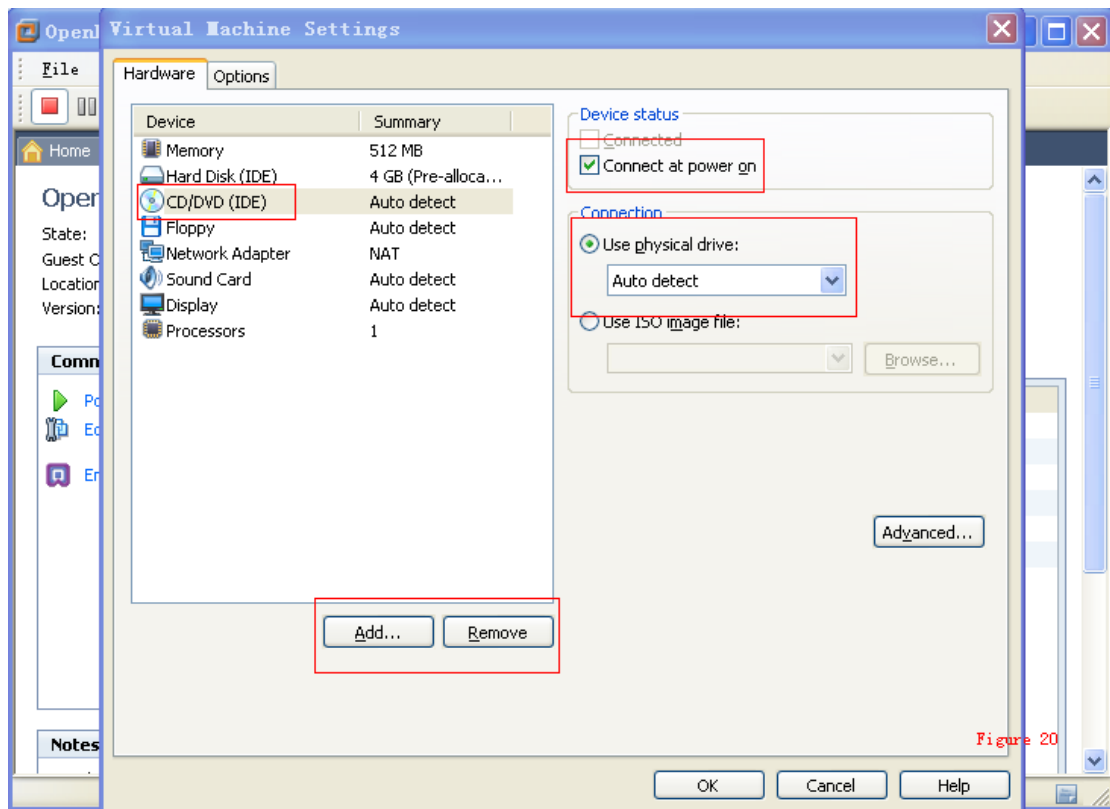


图 20

鼠标左键单击选择“Hardware”选项卡下面的“CD/DVD(IDE)”，就会出现右边的界面，在“Device

Status”（“设备状态”）处选择“Connect at power on”（“开机连接”）；在“Connection”下选择“Use physical drive”（“使用物理驱动器”，并设置为“Auto detect”（“自动检测”），此处如果对于没有安装虚拟光驱的朋友来说就选择“Use ISO image file”（“使用 ISO 镜像文件”），并浏览“Browse”你硬盘上 OpenBSD 的 install4.5.iso 文件所在位置，选中。图 20 中还有一个“Add”（“添加”）和一个“Remove”（“移除”）按钮，这就是用来添加或者移除虚拟硬件设备用的。

★★★★★特别说明：在创建玩虚拟磁盘后，因为全是碎片文件，最好先对硬盘做一个碎片整理，然后再开始安装 OpenBSD。在安装完 OpenBSD 后，做一个 Snapshot（快照，相当于 windows 中的 ghost，一键还原）就磁盘碎片整理一次，根据我的经验，如果今后一次性整理的话，基本上就无法整理了。硬盘上全是碎片文件看起来很不爽是不是？^_^★★★★★

在安装 OpenBSD 应该先做一下虚拟磁盘的规划，最好先拿一支笔和一张纸记录下来。因为有 4G 空间，我们作如下规划：

◆wd0a/ root 分区，256MB；

◆wd0b/(swap)交换分区, 256MB;

◆wd0d/tmp 临时文件分区, 256MB;

◆wd0e/var 256MB,如果要做 web 服务器或者邮件服务器的话, 这个区就要大一些, 因为数据库都安装在这个去。

◆wd0f /usr 用户区, 2GB。这个区要很大, 因为 OpenBSD 安装后, 它的基本文件都在这个区, 以后添加其他软件也都安装在这个区。

◆wd0g/home 家分区 (这样翻译真难听-_-), 1GB。

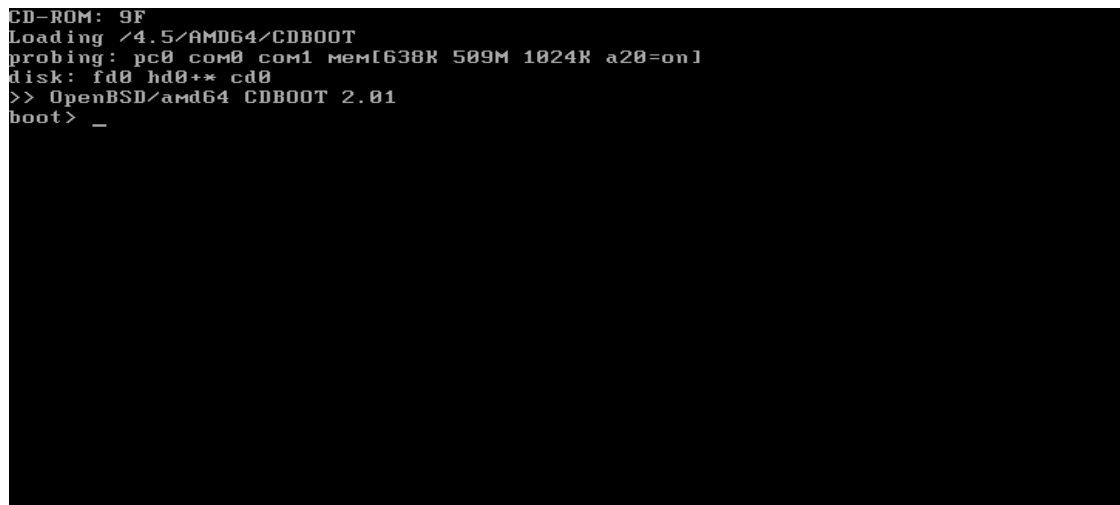
※注: 上面的 wd0a,wd0b 之类的名字可以先不管。只作简要说明, wd 表示 IDE 硬盘, 0 表示第一块硬盘, a,b,d 表示第一、二、三个分区。为什么没有 wd0c 呢? 这个去默认为安装 windows 程序系统使用的, OpenBSD 不能使用。

下面我们使用虚拟光驱加载 OpenBSD 的 ISO 镜像文件开始安装, 加载文件的方法就不用我说了吧。在加载了 ISO 文件后, 鼠标左键单击 “Power on this virtual machine” 按钮或者鼠标左键单击 VMware 菜单来 “Edit” 下面那个绿色的按钮, 打卡虚拟机。如图 21 所示:

※注: 从图 21 开始我将使用 VMware 的 Capture Screen 功能截图。

启动虚拟机, 加载安装介质, 开始安装

参见图 21

A screenshot of a terminal window showing the OpenBSD CDBOOT boot process. The text is as follows:

```
CD-ROM: 9F
Loading /4.5/AMD64/CDBOOT
probing: pc0 com0 com1 mem[638K 509M 1024K a20=on]
disk: fd0 hd0+* cd0
>> OpenBSD/amd64 CDBOOT 2.01
boot> _
```

图 21

这里回暂停 5s, 可以直接按回车, 或者让它自己继续运行, 后面我回提到。下面见图 22:

```

"Intel 82371AB Power" rev 0x00 at pci0 dev 7 function 3 not configured
vga1 at pci0 dev 15 function 0 "VMware Virtual SUGA II" rev 0x00
wsdisplay0 at vga1 mux 1: console (80x25, vt100 emulation)
mpi0 at pci0 dev 16 function 0 "Symbios Logic 53c1030" rev 0x01: apic 1 int 17 (
irq 9)
scsibus1 at mpi0: 16 targets, initiator 7
ppb1 at pci0 dev 17 function 0 "VMware Virtual PCI-PCI" rev 0x02
pci2 at ppb1 bus 2
em0 at pci2 dev 0 function 0 "Intel PRO/1000MT (82545EM)" rev 0x01: apic 1 int 1
8 (irq 11), address 00:0c:29:cd:41:f3
"Ensoniq AudioPCI97" rev 0x02 at pci2 dev 1 function 0 not configured
Vendor "VMware", unknown product 0x0740 (class system subclass miscellaneous, re
v 0x10) at pci2 dev 2 function 0 not configured
isa0 at mainbus0
com0 at isa0 port 0x3f8/8 irq 4: ns16550a, 16 byte fifo
com1 at isa0 port 0x2f8/8 irq 3: ns16550a, 16 byte fifo
pckbc0 at isa0 port 0x60/5
pckbd0 at pckbc0 (kbd slot)
pckbc0: using irq 1 for kbd slot
wskbd0 at pckbd0: console keyboard, using wsdisplay0
rd0: fixed, 4480 blocks
softraid0 at root
root on rd0a swap on rd0b dump on rd0b
Erase ^?, werase ^W, kill ^U, intr ^C, status ^T
(I)nstall, (U)pgrade or (S)hell? i_

```

图 22

完成初始化后进入安装界面，

Root on rd0a swap on rd0b dump on rd0b

Erase ^?, werase ^W, kill ^U, intr ^C, status ^T

(I)nstall, (U)pgrade or (S)hell? i **Enter**

输入 i 后，按 Enter 键确认，进入下一步，见图 23

```

system in an inconsistent state.

Terminal type? [vt220]
kbd(8) mapping? ('L' for list) [none] L
Major tables: be br cf de dk es fr hu it jp la lt lv nl no pl pt ru sf sg si sv
tr ua uk us
kbd(8) mapping? ('L' for list) [none] us
kbd: keyboard mapping set to us

IS YOUR DATA BACKED UP? As with anything that modifies disk contents, this
program can cause SIGNIFICANT data loss.

It is often helpful to have the installation notes handy. For complex disk
configurations, relevant disk hardware manuals and a calculator are useful.

Proceed with install? [no] yes
Cool! Let's get to it.

You will now initialize the disk(s) that OpenBSD will use. To enable all
available security features you should configure the disk(s) to allow the
creation of separate filesystems for /, /tmp, /var, /usr, and /home.

Available disks are: wd0.
Which one is the root disk? (or 'done') [wd0]
Do you want to use *all* of wd0 for OpenBSD? [no] yes_

```

图 23

为了节省篇幅，能在一幅图里显示的我就放在一幅图里，这样减少了图片的使用。

这一步会选择终端类型 (Terminal type) [vt220] 是默认的，直接回车确认；

接着是 kbd(8) mapping? ("L" for list) [none] (kbd 就是 keyboard，(8) 是在 manual

手册里的所属部分，不用管，这里就是选择键盘映射模式），默认的为[none],可以直接回车，因为我们中国都是用的美式键盘，不会有问题的。这里用列出所有国家的键盘映射模式，我选择了一个us,然后回车确认；

然后在 “Proceed with install?[no]” 后输入 yes 或者 y,回车确认开始安装；

然后选择可用的磁盘[wd0],回车确认；

然后问是否使用 wd0 整块磁盘来安装 OpenBSD，默认为[no],如果您还要在此虚拟磁盘上安装其他系统的话就回车。这里我们选择整块磁盘，输入 yes，回车确认；

下面进入图 24 所示内容，开始设置磁盘（Setting up disks）

```
c [part] - change partition size      r - display free space
D - reset label to default          s [path] - save label to file
d [part] - delete partition          u - undo last change
e - edit drive parameters           w - write label to disk
g [d : u] - [disk or [ulser geometry X - toggle expert mode
l [unit] - print disk label header   x - exit & lose changes
M - disklabel(8) man page           z - delete all partitions
m [part] - modify partition

Suffixes can be used to indicate units other than sectors:
    'b' (bytes), 'k' (kilobytes), 'm' (megabytes), 'g' (gigabytes)
    'c' (cylinders), '%' (% of total disk), '&' (% of free space).
Values in non-sector units are truncated to the nearest cylinder boundary.
> a a
offset: [63]
size: [8385867] 256M
Rounding to cylinder: 530082
FS type: [4.2BSD]
Mount point: [none] /
> a b
offset: [530145]
size: [7855785] 256M
Rounding to cylinder: 530145
FS type: [swap]
> _
```

图 24

在提示符 “>” 后输入 help,回车确认，会显示图上方的磁盘设置所有可用的命令。

然后在提示符 “>” 后输入 a a,前面一个 a 表示创建磁盘分区的命令，后面一个 a 是表示磁盘分区序列；偏移量 (offset) 不用管，在大小 (size) 后面输入大小 256m,m 表示 MB，G 表示 GB，这里不区分大小写。FS type 是文件系统 (file system) 的意思，默认[4.2BSD],回车确认就是，其他不管。在挂载点 (mount point) 处输入 / 表示根分区。后面的照着我们在前面部分规划的输入。创建 b 分区的时候文件系统类型是默认交换分区 (swap)，回车确认就是，不要去修改。

下面进入图 25:

```

FS type: [4.2BSD]
Mount point: [none] /var
> a f
offset: [2120580]
size: [6265350] 2g
Rounding to cylinder: 4209030
FS type: [4.2BSD]
Mount point: [none] /usr
> a g
offset: [6329610]
size: [2056320]
FS type: [4.2BSD]
Mount point: [none] /home
> p m
OpenBSD area: 0.0M-4094.7M; size: 4094.7M; free: 0.0M
#
  size      offset  fstype  [fsize bsize  cpg]
a:    258.8M      0.0M   4.2BSD   2048 16384    1 # /
b:    258.9M    258.9M   swap
c:   4096.0M      0.0M  unused
d:    258.9M    517.7M   4.2BSD   2048 16384    1 # /tmp
e:    258.9M    776.6M   4.2BSD   2048 16384    1 # /var
f:   2055.2M   1035.4M   4.2BSD   2048 16384    1 # /usr
g:    1004.1M   3090.6M   4.2BSD   2048 16384    1 # /home
> w
> q_

```

图 25

分区做好后，输入 p m 显示分区情况，然后输入 w,回车，q，再回车。w 表示写分区表，q 表示退出。

下面进入图 26

```

b:    258.9M    258.9M   swap
c:   4096.0M      0.0M  unused
d:    258.9M    517.7M   4.2BSD   2048 16384    1 # /tmp
e:    258.9M    776.6M   4.2BSD   2048 16384    1 # /var
f:   2055.2M   1035.4M   4.2BSD   2048 16384    1 # /usr
g:    1004.1M   3090.6M   4.2BSD   2048 16384    1 # /home
> w
> q
No label changes.
Mount point for wd0d (271360 KBytes)? (or 'none' or 'done') [/tmp]
Mount point for wd0e (271360 KBytes)? (or 'none' or 'done') [/var]
Mount point for wd0f (2048 MBytes)? (or 'none' or 'done') [/usr]
Mount point for wd0g (1024 MBytes)? (or 'none' or 'done') [/home]
Mount point for wd0d (271360 KBytes)? (or 'none' or 'done') [/tmp] done
No more disks to initialize.

OpenBSD filesystems:
wd0a /
wd0d /tmp
wd0e /var
wd0f /usr
wd0g /home

The next step *DESTROYS* all existing data on these partitions!
Are you really sure that you're ready to proceed? [no] yes_

```

图 26

上图中间部分有个确认，一直按回车，之道出现循环[/tmp]后输入 done，回车确认。

然后开始下一步，输入 yes，回车确认。进入图 27:

```

on May  4 21:40:21 2009)
/dev/wd0d on /mnt/tmp type ffs (rw, asynchronous, local, nodev, nosuid, ctime=Mon May  4 21:40:21 2009)
/dev/wd0f on /mnt/usr type ffs (rw, asynchronous, local, nodev, ctime=Mon May  4 21:40:21 2009)
/dev/wd0e on /mnt/var type ffs (rw, asynchronous, local, nodev, nosuid, ctime=Mon May  4 21:40:21 2009)

System hostname? (short form, e.g. 'foo') xiechangjun
Configure the network? [yes]
Available interfaces are: em0.
Which one do you wish to initialize? (or 'done') [em0]
Symbolic (host) name for em0? [xiechangjun]
The media options for em0 are currently
    media: Ethernet autoselect (100baseT full-duplex, master)
Do you want to change the media options? [no]
IPv4 address for em0? (or 'none' or 'dhcp') [dhcp]
Issuing hostname-associated DHCP request for em0.
DHCPDISCOVER on em0 to 255.255.255.255 port 67 interval 1
DHCPDISCOVER on em0 to 255.255.255.255 port 67 interval 2
DHCPOFFER from 192.168.150.254
DHCPREQUEST on em0 to 255.255.255.255 port 67
DHCPACK from 192.168.150.254
bound to 192.168.150.130 -- renewal in 900 seconds.
IPv6 address for em0? (or 'rtsol' or 'none') [none] _

```

图 27

见上图 27, 开始设置系统主机名 (Setting system hostname), 随便您, 这里我们输入: xiechangjun, 回车确认;

下一步问是否配置网络 (Config the network?[yes]) 默认为是, 直接回车确认;

然后出现可用的网络接口卡 em0, 并问初始化那一块网卡, 因为这里只有 em0, 所以直接回车确认。这里, 如果您虚拟了两块网络接口卡的话, 它会显示出来, 您随便选择一块可用的就行;

后面一直回车到 (IPv4 address for em0?(or 'none' or 'dhcp') [dhcp], 因为是虚拟机, 采用 DHCP 获得 IP 最方便, 所以我们选择使用 DHCP 方式获得 IP), 直接回车确认。然后您就会看到网卡 em0 获取了 IP, 并绑定到 192.168.150.130。关于 DHCP 的原理请查看其他资料或者利用丰富的网络资源, 在此不再赘述。

下一步询问 IPv6 的设置, 这里不使用, 全部根据默认[no]的设置直接回车确认就行。

进入图 28:

```

Do you want to change the media options? [no]
IPv4 address for em0? (or 'none' or 'dhcp') [dhcp]
Issuing hostname-associated DHCP request for em0.
DHCPDISCOVER on em0 to 255.255.255.255 port 67 interval 1
DHCPDISCOVER on em0 to 255.255.255.255 port 67 interval 2
DHCPOFFER from 192.168.150.254
DHCPREQUEST on em0 to 255.255.255.255 port 67
DHCPACK from 192.168.150.254
bound to 192.168.150.130 -- renewal in 900 seconds.
IPv6 address for em0? (or 'rtsol' or 'none') [none]
No more interfaces to initialize.
DNS domain name? (e.g. 'bar.com') [localdomain]
DNS nameserver? (IP address or 'none') [192.168.150.2]
Use the nameserver now? [yes]
Default IPv4 route? (IPv4 address, 'dhcp' or 'none') [dhcp]
Edit hosts with ed? [no]
Do you want to do any manual network configuration? [no]
Password for root account? (will not echo)
Password for root account? (again)

Let's install the sets!
Location of sets? (cd disk ftp http or 'done') [cd]
Available CD-ROMs are: cd0.
Which one contains the install media? (or 'done') [cd0]
Pathname to the sets? (or 'done') [4.5/amd64] _

```

图 28

如上图 28 所示，依次回车一直到密码设置 (Password for root account?(will not echo))，就是为 root 账户设置密码，虽然是虚拟机，但是为了养成好的系统管理习惯，还是要设置密码，注意，这里您输入的密码不会显示任何内容。当您输入完后，回车确认，然后进一步确认输入密码，以保证两次输入的一致性，然后回车确认。

开始选择安装介质，根据我们此次实验所设置的环境，加载介质 (Location of sets?(cd disk ftp http or 'done') [cd])，我们直接回车确认，后面依次回车。

下面进入图 29：

```
Let's install the sets!
Location of sets? (cd disk ftp http or 'done') [cd]
Available CD-ROMs are: cd0.
Which one contains the install media? (or 'done') [cd0]
Pathname to the sets? (or 'done') [/4.5/amd64]

Select sets by entering a set name, a file name pattern or 'all'. De-select
sets by prepending a '-' to the set name, file name pattern or 'all'. Selected
sets are labelled '[X]'.

[X] bsd
[X] bsd.rd
[ ] bsd.mp
[X] base45.tgz
[X] etc45.tgz
[X] misc45.tgz
[X] comp45.tgz
[X] man45.tgz
[X] game45.tgz
[ ] xbase45.tgz
[ ] xetc45.tgz
[ ] xshare45.tgz
[ ] xfont45.tgz
[ ] xserv45.tgz
Set name? (or 'done') [bsd.mp] all_
```

图 29

见上图 29，Set name? (or 'done') [bsd.mp]，这一步询问我要选择安装那些组件，我们先选择所有，输入 all，回车确认，进入图 30：

```
[X] comp45.tgz
[X] man45.tgz
[X] game45.tgz
[ ] xbase45.tgz
[ ] xetc45.tgz
[ ] xshare45.tgz
[ ] xfont45.tgz
[ ] xserv45.tgz
Set name? (or 'done') [bsd.mp] all

[X] bsd
[X] bsd.rd
[X] bsd.mp
[X] base45.tgz
[X] etc45.tgz
[X] misc45.tgz
[X] comp45.tgz
[X] man45.tgz
[X] game45.tgz
[X] xbase45.tgz
[X] xetc45.tgz
[X] xshare45.tgz
[X] xfont45.tgz
[X] xserv45.tgz
Set name? (or 'done') [done] -bsd.mp_
```

图 30

是不是所有的组件都选上了？因为我的笔记本是单核的，而 bsd.mp 那个文件支持多核计算机

用的，在这里用不着，输入 `-bsd.mp`，回车确认。这里，需要添加什么就输入“+文件名”，不需要什么的时候就输入“-文件名”。就如前面的“-bsd.mp”，如果要加上的话就为“+bsd.mp”。

至于各个组件的作用请见 <http://www.openbsd.org/faq/faq4.html>，这里不赘述。

进入图 31：

```
[X] comp45.tgz
[X] man45.tgz
[X] game45.tgz
[X] xbase45.tgz
[X] xetc45.tgz
[X] xshare45.tgz
[X] xfont45.tgz
[X] xserv45.tgz
Set name? (or 'done') [done] -bsd.mp

[X] bsd
[X] bsd.rd
[ ] bsd.mp
[X] base45.tgz
[X] etc45.tgz
[X] misc45.tgz
[X] comp45.tgz
[X] man45.tgz
[X] game45.tgz
[X] xbase45.tgz
[X] xetc45.tgz
[X] xshare45.tgz
[X] xfont45.tgz
[X] xserv45.tgz
Set name? (or 'done') [bsd.mp] done_
```

图 31

在做好最终选择后，输入 `done`，回车确认。

进入图 32：

```
Getting comp45.tgz ...
100% |*****| 98 MB 01:38
Getting man45.tgz ...
100% |*****| 8032 KB 00:09
Getting game45.tgz ...
100% |*****| 2642 KB 00:02
Getting xbase45.tgz ...
100% |*****| 11002 KB 00:10
Getting xetc45.tgz ...
100% |*****| 78070 00:00
Getting xshare45.tgz ...
100% |*****| 2697 KB 00:05
Getting xfont45.tgz ...
100% |*****| 34745 KB 00:29
Getting xserv45.tgz ...
100% |*****| 20987 KB 00:18
Location of sets? (cd disk ftp http or 'done') [done]
Start sshd(8) by default? [yes]
Start ntpd(8) by default? [no] yes
NTP server? (hostname or 'default') [default]
Do you expect to run the X Window System? [no] yes
Change the default console to com0? [no]
Saving configuration files...done.
Generating initial host.random file...done.
What timezone are you in? ('?' for list) [Canada/Mountain] _
```

图 32

上图 32 的上面部分就是安装的过程，自动的，不用管，看看就行了。^_^安装完成后又有一个 Location of sets (cd disk ftp http or ‘done’) [done]，因为不再有其他需要，直接回车确认。

后面根据您的需要, 输入 yes 或者 no, 确认就是。如果今后要使用图形界面的话, 设置 X Window System 那里必须是 yes。

然后设置时区 (What timezone are you in? ('?' for list), 默认为 [Canada/Mountain], 我们在中国, 肯定得另外设置, 输入 ?, 回车确认, 进入图 33:

```
Getting xfont45.tgz ...
100% |*****! 34745 KB    00:29
Getting xserv45.tgz ...
100% |*****! 20987 KB    00:18
Location of sets? (cd disk ftp http or 'done') [done]
Start sshd(8) by default? [yes]
Start ntpd(8) by default? [no] yes
NTP server? (hostname or 'default') [default]
Do you expect to run the X Window System? [no] yes
Change the default console to com0? [no]
Saving configuration files...done.
Generating initial host.random file...done.
What timezone are you in? ('?' for list) [Canada/Mountain] ?
Africa/      Chile/      GB-Eire      Israel       NZ-CHAT      UCT
America/     Cuba/       GMT          Jamaica      Navajo       US/
Antarctica/  EET         GMT+0       Japan        PRC          UTC
Arctic/      EST         GMT-0       Kwajalein    PST8PDT      Universal
Asia/        EST5EDT     GMT0        Libya        Pacific/     W-SU
Atlantic/    Egypt      Greenwich   MET          Poland       WET
Australia/   Eire       HST         MST          Portugal     Zulu
Brazil/      Etc/       Hongkong    MST7MDT      ROC          posix/
CET          Europe/    Iceland     Mexico/      ROK          posixrules
CST6CDT      Factory   Indian/     Mideast/     Singapore    right/
Canada/      GB         Iran        NZ           Turkey       zone.tab
What timezone are you in? ('?' for list) [Canada/Mountain] Asia/_
```

图 33

根据实际情况, 我们在最后那里输入 Asia/, 回车确认。下面进入图 34:

```
Bishkek      Irkutsk      Magadan      Saigon      Uientiane
Brunei       Istanbul     Makassar     Sakhalin    Vladivostok
Calcutta     Jakarta     Manila       Samarkand   Yakutsk
Choibalsan   Jayapura    Muscat       Seoul       Yekaterinburg
Chongqing    Jerusalem    Nicosia     Shanghai    Yerevan
What sub-timezone of 'Asia/' are you in? ('?' for list) Chongqing
Setting local timezone to 'Asia//Chongqing'...done.
Making all device nodes...done.
Installing boot block...
boot: /mnt/boot
proto: /usr/mdcc/biosboot
device: /dev/rwd0c
/usr/mdcc/biosboot: entry point 0
proto bootblock size 512
/mnt/boot is 3 blocks x 16384 bytes
fs block shift 2; part offset 63; inode block 160, offset 12456
using MBR partition 3: type 166 (0xa6) offset 63 (0x3f)
done.

CONGRATULATIONS! Your OpenBSD install has been successfully completed!
To boot the new system, enter halt at the command prompt. Once the
system has halted, reset the machine and boot from the disk.
# reboot
syncing disks... done
```

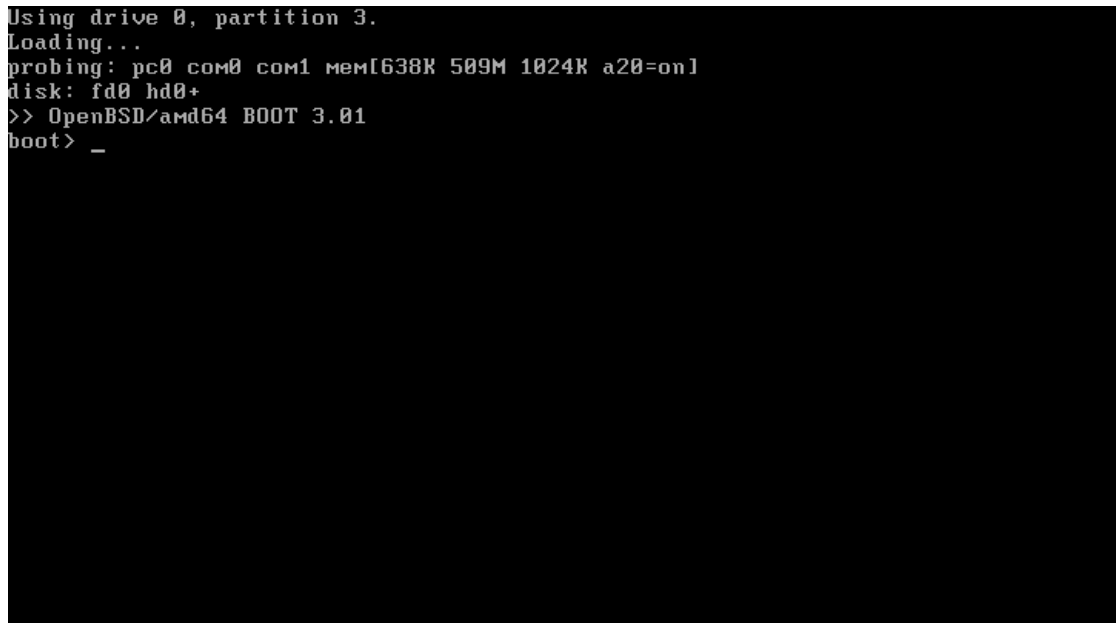
图 34

根据情况, 输入 Chongqing, 回车确认, 系统提示时区设置好了 (Setting local timezone to 'Asia//Chongqing' ...done.)。

到此步骤就算是安装完成了了。最后需要重启系统, 输入 reboot 或者 halt 都行, 然后回车确

认。大功告成！

OpenBSD 的基本设置即操作，见图 35：



```
Using drive 0, partition 3.
Loading...
probing: pc0 com0 com1 mem[638K 509M 1024K a20=on]
disk: fd0 hd0+
>> OpenBSD/amd64 BOOT 3.01
boot> _
```

图 35

见上图 35，这是重启后开机的第一个画面，我们可以看见有一个“boot>”提示符，系统会在那里暂停 5s（默认时间长短，可以自己修改），这里是让我们选择登陆模式，进入单用户模式就从这一步开始，在“boot>”后面输入 boot -s，表示进入单用户模式。在这里我们先不管，只是介绍一个进入单用户模式的方法。直接回车就是，或者等待 5s 让系统自己启动。

下面我们看图 36：



```
DHCPREQUEST on em0 to 255.255.255.255 port 67
DHCPACK from 192.168.150.254
bound to 192.168.150.130 -- renewal in 900 seconds.
starting system logger
starting initial daemons: ntpd.
savecore: no core dump
checking quotas: done.
building ps databases: kvm dev.
clearing /tmp
starting pre-securelevel daemons:.
setting kernel security level: kern.securelevel: 0 -> 1
creating runtime link editor directory cache.
preserving editor files.
ssh-keygen: generating new DSA host key... done.
ssh-keygen: generating new RSA host key... done.
ssh-keygen: generating new RSA1 host key... done.
openssl: generating new isakmpd RSA key... done.
starting network daemons: sendmail inetd sshd.
starting local daemons:.
standard daemons: cron.
Tue May 5 05:49:46 CST 2009

OpenBSD/amd64 (xiechangjun.localdomain) (ttyC0)
login: _
```

图 36

图 36 已经进入到登陆界面了。在 login:后面输入用户名 root，回车确认，然后输入我们在前面

设置的密码，回车确认就可以登陆了。

※注意：这里只是实验环境，为了养成良好的安全习惯，建议以后新建一个常规账户登陆，需要使用超级用户(root)权限的时候再用 su 命令进入超级用户，使用完后用 exit 命令退出超级用户回到常规用户。

进入图 37：

```
Password:
OpenBSD 4.5 (GENERIC) #2052: Sat Feb 28 14:55:24 MST 2009

Welcome to OpenBSD: The proactively secure Unix-like operating system.

Please use the sendbug(1) utility to report bugs in the system.
Before reporting a bug, please try to reproduce it with the latest
version of the code. With bug reports, please try to ensure that
enough information to reproduce the problem is enclosed, and if a
known fix for it exists, include that as well.

You have mail.
Terminal type? [vt220]
# df -h
Filesystem      Size   Used  Avail Capacity  Mounted on
/dev/wd0a        254M   38.5M   202M     16%      /
/dev/wd0g        988M    2.0K    939M      0%     /home
/dev/wd0d        254M    6.0K    241M      0%     /tmp
/dev/wd0f        2.0G   695M    1.2G     36%     /usr
/dev/wd0e        254M    4.1M    237M      2%     /var
# date
Tue May  5 05:51:19 CST 2009
# rdate -n time.windows.com
Mon May  4 21:51:57 CST 2009
# ftp ftp.openbsd.org_
```

图 37

成功登陆系统后会有一个欢迎界面。在选择终端类型（Terminal Type?[vt220]）那里直接回车确认即可。

※注：此时的命令提示符是“#”，表明是以 root 角色登陆的，如果是普通角色登陆的话，命令提示符就会是“\$”。

下面我们使用 df -h 命令查看一下当前磁盘使用情况，以验证我在前面说的话。df 是 disk free 的意思，-h 选项是“Human-readable” output 的意思。其他选项可以使用 OpenBSD 系统自带的 manual 手册。在命令提示如“#”后面输入 man df，回车确认即可。显示页面支持键盘上的 PageUp、PageDown 键。

因为 OpenBSD 默认没有开启 NTPD（Network Time Protocol daemon）实用程序同步网络更新本地系统日期时间。所以日期时间会不准确。使用 date 命令查看当前系统日期时间，明显不对。那么我们就势必想更新日期时间。使用 rdate -n time.windows.com，回车确认即可，更新成功后会如图 37 所示显示处正确的日期时间。rdate(set the system's date from a remote host)，即从远程主机设置本地系统的日期。

再简单介绍一下 FTP 的使用，仍如图 37 所示，在命令提示符“#”后面输入：

ftp ftp.openbsd.org 回车确认。进入图 38：

```
Welcome to OpenBSD: The proactively secure Unix-like operating system.

Please use the sendbug(1) utility to report bugs in the system.
Before reporting a bug, please try to reproduce it with the latest
version of the code.  With bug reports, please try to ensure that
enough information to reproduce the problem is enclosed, and if a
known fix for it exists, include that as well.

You have mail.
Terminal type? [vt220]
# df -h
Filesystem      Size    Used    Avail Capacity  Mounted on
/dev/wd0a       254M    38.5M    202M     16%      /
/dev/wd0g       988M     2.0K    939M      0%      /home
/dev/wd0d       254M     6.0K    241M      0%      /tmp
/dev/wd0f       2.0G    695M    1.2G     36%      /usr
/dev/wd0e       254M     4.1M    237M      2%      /var
# date
Tue May  5 05:51:19 CST 2009
# rdate -n time.windows.com
Mon May  4 21:51:57 CST 2009
# ftp ftp.openbsd.org
Connected to openbsd.sunsite.ualberta.ca.
220 openbsd.srv.ualberta.ca FTP server ready.
Name (ftp.openbsd.org:root): ftp_
```

图 38

成功连接远程 FTP 服务器后，会如图 38 所示提示输入用户名（Name([ftp.openbsd.org:root](ftp://ftp.openbsd.org)):），

在后面键入用户名 ftp 或者 anonymous,回车确认即可,这是匿名用户登陆,密码为空,直接回车确认。关于 FTP 使用的详细教程请查询互联网资料。我这里给出这个例子只是给初学者一个感官上的认识。下面进入图 39:

```

230-      CD sales are important to support the
230-      continued development of the project.
230-
230-      UK
230-
230-      *DO NOT* mirror openbsd from this site! use one of the
230-      "second level mirrors" listed at http://www.openbsd.org/ftp.html
230-      instead of this site. If you mirror from this site you will lose
230-      access to it.
230-
230-      E-mail comments, questions, trouble reports, and complaints
230-      to beck@openbsd.org. Please drive safely.
230-
230-      Guest login ok, access restrictions apply.
230-      Remote system type is UNIX.
230-      Using binary mode to transfer files.
230-      ftp> close
230-      221 Goodbye.
230-      ftp> bye
230-      #

```

图 39

如图 39 所示，登陆远程 FTP 服务器成功后，会有一个欢迎画面。由于这不是本教程的主题，所以我们先断开连接，退出 FTP，进入下一个环节。使用 `close` 命令关闭连接，`exit` 或者 `bye` 命令退出 FTP 服务器。


```
ftp> bye
# export PKG_PATH=ftp://ftp.openbsd.org/pub/OpenBSD/4.5/packages/amd64/
# pkg_add -v bzip2-1.0.5.tgz
Error from ftp://ftp.openbsd.org/pub/OpenBSD/4.5/packages/amd64/bzip2-1.0.5.tgz
:
ftp: connect: Connection refused
ftp: Can't connect or login to host 'ftp.openbsd.org'
parsing bzip2-1.0.5.tgz
found libspec c.50.1 in /usr/lib
bzip2-1.0.5: complete
# shutdown -ph now
Shutdown NOW?
shutdown: [pid 31271]
#

*** FINAL System shutdown message from root@xiechangjun.localdomain ***
System going down IMMEDIATELY

System shutdown time has arrived
/etc/rc.shutdown in progress...
/etc/rc.shutdown complete.
#
```

图 40

下面简单介绍一下使用 pkg_add 命令安装程序的过程，该过程如图 40 所示。注意有大小写区分噢！

```
#export PKG_PATH=ftp://ftp.openbsd.org/pub/OpenBSD/4.5/packages/amd64/ ①
```

```
#pkg_add -v bzip2-1.0.5.tgz ②
```

上面①是设置 pkg_add 命令将要安装的文件所在的远程服务器路径；②就是使用 pkg_add 开始安装，-v (verbose) 选项是显示详细过程的意思。后面出现了一些错误，是我的网络问题，不过正好借助这个问题说明 pkg_add 使用程序会反复试探网络连接，直到安装成功为止。关于 pkg_add 使用程序的更多介绍，请查看 OpenBSD 系统自带的 manual 手册。查看方法：man pkg_add，回车确认。

到此，关于 OpenBSD 的基本安装以及简单的介绍就结束了。相信读者已经大致了解了 OpenBSD 的安装过程。要想熟悉整个过程，最关键的一步还是自己动手，亲自安装，我只是在这里做一个引导，也算是为初学者做了一点点贡献吧，献丑了 O(n_n)O~最后说一句：学习，动手实践是关键。勤使用 manual 手册，多 Google！

网友热评

热点技术评论

[extern "C" 是干吗用的?](#)
[高手来解决下 求一个高难度正则表达式](#)
[《c 专家编程》中一个声明讲解没看懂](#)
[大家用 CommVault 备份的软件多么](#)
[最精简的 QuickSort 快速排序?](#)
[世界 500 强软件公司的算法笔试题](#)
[TCP 服务端最多可和多少客户端同时保持连接?](#)
[某块内存, 不能赋值会是什么原因](#)
[请问: UDP 和 TCP 包中的数据在什么地方](#)
[如何让 c 语言嵌入 .swf 动画文件?](#)
[找数列规律的问题, 能写出 C 的算法吗?](#)
[socket 代码中获得由内核自动分配端口号](#)
[类的大小在什么情况下为 0?](#)
[linux 内核调试的郁闷历程!!!](#)
[为什么说作开发一般不使用 root 帐号?](#)
[为什么我输入 ./configure 现实没有文件夹](#)
[一个简单文件系统的实现](#)
[cacti 问题](#)
[xp 无法访问 samba 服务器](#)
[如何加快文件 copy 速度?](#)
[怎么让 redhat 企业版的改成 centos 5.3?](#)
[直接建 swap 分区和建立 swapfile 有何不同?](#)
[基于 CentOS 5.2 的 RHCS 的高可用性解决方案](#)
[RedHat 如何升级内核](#)
[不用锁实现队列读写?](#)
[Unix 平台下的程序大家使用什么开发环境](#)

热点新闻评论

[Unix/Linux 使用感觉调查](#)
[GCC 4.4.1 发布 C++ 0x 实验性支持大幅进步](#)
[反 java 大神产生了.....](#)
[北京百度招聘备份/分布式系统人才](#)
[搞 Linux 的哥们都在什么样的公司就职啊?](#)
[毛老出了本《windows 内核源代码情景分析》](#)
[CU 里有多少外企的?](#)
[做算法开发好累啊!](#)
[任正非论技术自主和拿来: 新开发超 30%浪费](#)
[Sun 存储业务令 Oracle 不忍弃之](#)
[经济危机下数据备份省钱之道](#)
[9 岁娃玩电脑堪比专家 熟悉十多种编程语言](#)
[联想首席研究员: 联想技术上完全领先同行](#)
[暴风影音导致国内 DNS 服务器瘫痪](#)
[c 程序员重复造轮子是否是好事](#)
[LINUX, 真的让我太失望了](#)
[WINDOWS, 真的让我太失望了](#)
[为什么说 Linux 还没有为桌面做好准备](#)
[Linux 有待完善的小缺陷的累记](#)
[北京今年的 Linux 行业形势咋样](#)
[大家做项目的时候, 最怕遇到什么问题?](#)
[三大特性!Linux 7 桌面正式版初体验\(组图\)](#)
[这本 C++ 的书翻译得怎么样?](#)
[统计一下, 经济危机中失业了再找工作的朋友](#)
[期待有人组织组织团购 OpenBSD 官方 T-Shirt](#)
[在多核系统上网络数据转发实验和一点思考](#)