

Linux 运维趋势

第六期

2011年3月号 关键字：FTP, rsync, crontab, dump, shell

备份



内容目录

【人物】

专访田逸：互联网运维领域的五年变迁.....3

【交流】

Stack Overflow 架构揭秘（软硬件、人员、数字）.....5

八卦，趣闻与数字 2011.2 – 2011.3.....9

【专题】

本期专题：备份.....10

Linux 下的开源数据备份和恢复软件.....11

Facebook 是怎么做 MySQL 备份的?.....15

Linux 中常用的 Shell 备份脚本三则.....17

Linux VPS/服务器上导入导出 MySQL 数据库.....21

自建基于 Linux 的 Time Capsule 服务.....22

【技巧】

Debian 下通过 Bonding 实现双网卡单 IP.....25

CentOS 5.5 下 rsync 使用技巧与权限问题解读.....26

Linux 运维趋势

杂志策划：[51CTO 系统频道](#)

本期主编：杨赛

封面制作：徐泽琼

交流圈子：

<http://g.51cto.com/linuxops>

专题页面邮件订阅入口：

<http://os.51cto.com/art/201011/233915.htm>

下载汇总页：

<http://down.51cto.com/zt/71>

投稿邮箱：

yangsai@51cto.com

你要往 CTO、CIO 的方向发展，就要去考虑这些问题，而不能局限在自己的技术领域当中。要有那个高度。

专访田逸：互联网运维领域的五年变迁

采访整理/杨赛



人物简介：

田逸 (Sery)，男，土家族，重庆秀山人，全真龙门 27 代俗家弟子。资深系统管理员，精通 Linux/Unix，多年系统管理经验，擅长并热衷于大规模的网络运维。公开发表 IT 相关的技术文章数十篇，并发布过《互联网运营智慧》一书。

他的博客：<http://sery.blog.51cto.com/>

田逸在互联网技术运营领域是一个自学成才的典范：中专毕业后分配到县缫丝厂做机修工，辗转多年历经多种职业后，于 2000 年投身 IT 行业。先后在中国科学院地质与地球物理研究所、国家旅游局信息中心、空中网从事网络和系统管理工作。独立运行维护节点数超过 1000、1500 多 windows 客户端、数百个

服务器的大规模网络。不过，更多人知道田逸 /sery 的名字，主要是因为田逸在技术社区中长期以来的热心分享和交流。很多年轻的系统管理员们都愿意在线上和线下请教田逸，并自称 sery 的徒弟。

2010 年底，田逸的第一本著作《互联网运营智慧》正式上架，他自己的工作内容也在过

去几个月有了一些变动。51CTO 系统频道在近日对田逸做了一次简短的线上+线下采访，跟大家分享田逸老师最新的成长心得。

51CTO：首先，能否请您简单介绍一下您最近三个月的工作重心和技术关注点？

田逸：我从去年 7 月开始单练，接了几个活，主要工作就是给人解决麻烦事情。一个客户的网站，访问量非常大（pv 千万吧），在境外租了 8 个服务器，经常被流量打死。接手过来后，就对其进行调整、优化，最后网站运行稳定，而且减少了 3 个服务器，省了一笔费用。

由于业务的特点，关注点主要集中在怎样选机房、怎么给客户节省费用及最大化使用资源上面。

51CTO：听说您其实也是半路出家进入 Linux 运维这个圈子的。您对这个转型的过程有什么特别印象深刻的事情？

田逸：进 IT 圈最初是修刻录机，后来公司垮了，有幸承包科学院某所的网络维护，才有机会接触 Linux，但基本上是小打小闹。大概

是 2005 年秋季的样子，我到空中网做系统管理员，被同事带到机房，哇塞！那么多服务器（好几百个）要我来管，心里有点打鼓，当时真没底气。

51CTO：就您个人的经历而言，您感觉五年前做运维，和现在做运维的最大区别是什么？

田逸：就互联网而言，变化确实是快。别说 5 年，就是一年前跟现在比，也有很大的不同。

五年前那阵子做运维，我坦白交代吧，应该还是比较粗糙的，也没有什么规划，也谈不上什么架构。那个部门要上项目，就是买机器，装系统，上架上线，最多再弄个监控，完事。

那么现在呢？得考虑规划、成本、资源分配、部门配合、故障快速处理、系统高可用可扩展性等等系列问题，不单单是技术问题了，因此我把它称之为技术运营。技术运营的好坏事关重大——比如选的机房不好，规划得不好，又费钱又不能提供良好的用户体验。

51CTO：您现在的主要修炼途径是什么？您觉得文档/书籍，相比与其他技术人员交流、

学习案例而言，哪种修炼途径更高效？

田逸：也没有什么主要的途径，有时看书、有时看网站。更有效的方式是边干边学。

51CTO：很多网友都对《互联网运营智慧》这本书的标题表示疑惑，觉得这明明是讲网站运维的书，为什么要取名叫做互联网运营智慧？您对于“运维”和“运营”是怎么看的？

田逸：其实我上面也说了，我一开始做运维，就是买机器，装系统，上架上线，最后再弄个监控。但是现在，我们需要考虑很多这些纯技术问题之外的方面：规划、成本、资源分配、部门配合、故障快速处理、系统高可用可扩展性等问题。我觉得这些都不单纯是“运维”一词所涵盖的，而是上升到了“技术运营”的层面。你要往 CTO、CIO 的方向发展，就要去考虑这些问题，而不能局限在自己的技术领域当中。要有那个高度。

51CTO：您对自己的《互联网运营智慧》这本书满意么？目前有没有下一步的写作计划？

田逸：《互联网运营智慧》是我写的第一本书，花了很多时间。在创作过程中，我一度

怀疑写不下去。每次我去书店，都鼓舞自己，一定要把书放到书店。还好，最后终于出版发行，还算是满意吧。

如果有能力和时间的话，还是会写的，但绝不去攒书。自己这方面，打算去翻译监控方面的书（nagios），但目前似乎出版界没人感兴趣。

51CTO：最后一个问题比较私人，但对于同行们也是很有指导性的：能否简单的描述一下您上一个工作日的 24 小时是怎样分配的？

田逸：其实我是一个时间比较混乱的人，也没有什么人生规划。就好像我写书一样，想到哪里写到哪里；先有内容，再有目录和章节的。■※

原文：
<http://os.51cto.com/art/201103/247256.htm>

相关阅读：
[运维人员应该掌握哪些常用技术](#)

[系统运维秘诀：变化，监控，扩展（技术篇）](#)

[漫谈运维：半神半仙亦民工](#)

<http://g.51cto.com/linuxops/>
<http://os.51cto.com/linux/>
投稿邮箱：yangsai@51cto.com

使用了 Proxy，而不是使用 Windows 网络负载均衡（NLB），因为 HAProxy 成本低廉、易于使用，还是免费的。

Stack Overflow 架构揭秘（软硬件、人员、数字）

文/Todd Hoff
编译/布加迪

Stack Overflow 是 IT 界最受欢迎的问答网站之一。自从本人在 2009 年写了关于 Stack Overflow 架构的第一篇文章后，该架构已发生了很大的变化。那篇文章主要介绍了 Stack Overflow 采取的向上扩展（scale-up）策略，而 Stack Overflow 在最近几年得到了全面发展。

Stack Overflow 取得了长足发展：规模扩大了一倍多，每月不重复的访问用户超过 1600 万；每月网页浏览量（PV）增长了近 6 倍，达到 9500 万。

Stack Overflow 发展壮大成了 Stack Exchange Network，而这个网络包括 Stack Overflow、Server Fault 和

Super User 等，旗下总共拥有 43 个网站，而且发展势头良好。

但不变的是 Stack Overflow 在其所作所为方面坚持的开放理念，而这才有了今天这篇文章。

这几年来比较明显的一些变化如下：

- ◆数量更多：更多的用户、更多的 PV、更多的数据中心、更多的站点、更多的开发人员更多的操作系统、更多的数据库、更多的机器。

- ◆Linux：Stack Overflow 因使用 Windows 系列产品而著称，现在他们使用越来越多的 Linux 机器，用于 HAProxy、Redis、Bacula、Nagios、日志和路由器等系统。所有支持功能似乎都由

Linux 来处理，这就需要开发并行版本发行流程。

- ◆容错：现在为 Stack Overflow 提供服务的是使用两条不同互联网连接的两只不同交换机，Stack Overflow 添加了冗余机器，一些功能已搬迁到第二个数据中心。

- ◆NoSQL：Redis 现用作整个网络的缓存层以前没有独立的缓存层，所以这是一大变化，使用基于 Linux 的 NoSQL 数据库也是一大变化。

下面是收集的一些信息：

统计数字

- ◆每月 PV 9500 万次

- ◆每秒 800 个 HTTP 请求

- ◆每秒 180 个 DNS 请求

- ◆每秒 55 Mbps

- ◆1600 万个用户——Stack Overflow 的流量在 2010 年增长了 131%，全球每月不重复访客增至 1660 万人。

数据中心

Stack Overflow 网络架构

- ◆ 1 个机架放在俄勒冈州的 Peak Internet (用于放置 chat 和 Data Explorer)

- ◆ 2 个机架放在纽约州的 Peer 1 (用于放置 Stack Exchange Network 的其余部分)

硬件

10 台戴尔 R610 IIS Web 服务器 (3 台专门用于 Stack Overflow)

- ◆ 1 个英特尔至强处理器 E5640 , 2.66 GHz 四核 , 8 线程
- ◆ 16 GB 内存
- ◆ Windows Server 2008 R2

2 台戴尔 R710 数据库服务器 :

- ◆ 2 个英特尔至强处理器 X5680 , 3.33 GHz
- ◆ 64 GB 内存
- ◆ 8 个硬盘
- ◆ SQL Server 2008 R2

2 台戴尔 R610 HAProxy 服务器 :

- ◆ 1 个英特尔至强处理器 E5640 , 2.66 GHz
- ◆ 4 GB 内存
- ◆ Ubuntu Server

2 台戴尔 R610 Redis 服务器 :

- ◆ 2 个英特尔至强处理器 E5640 , 2.66 GHz
- ◆ 16 GB 内存
- ◆ CentOS

1 台戴尔 R610 Linux 备份服务器 , 运行 Bacula :

- ◆ 1 个英特尔至强处理器 E5640 , 2.66 GHz
- ◆ 32 GB 内存

1 台戴尔 R610 Linux 管理服务器 , 用于 Nagios 和日志 :

- ◆ 1 个英特尔至强处理器 E5640 , 2.66 GHz
- ◆ 32 GB 内存

2 个戴尔 R610 VMWare ESXi 域控制器 :

- ◆ 1 个英特尔至强处理器 E5640 , 2.66 GHz

- ◆ 16 GB 内存

2 台 Linux 路由器

5 台戴尔 Power Connect 交换机

开发工具

- ◆ 编程语言 : C#
- ◆ IDE : Visual Studio 2010 团队套件
- ◆ 框架 : 微软 ASP.NET (版本 4.0)
- ◆ Web 框架 : ASP.NET MVC 3
- ◆ 视图引擎 : Razor
- ◆ 浏览器框架 : jQuery 1.4.2
- ◆ 数据访问层 : LINQ to SQL , 一些原始 SQL
- ◆ 源码控制 : Mercurial 和 Kiln
- ◆ 比较工具 : Beyond Compare 3

使用的软件和技术

- ◆ Stack Overflow 通过 BizSpark 使用 WISC 堆栈
- ◆ 操作系统 : Windows Server 2008 R2

- ◆数据库：运行微软 Windows Server 2008 企业版 x64 的 SQL Server 2008 R2
- ◆Ubuntu Server
- ◆CentOS
- ◆Web 服务器：IIS 7.0
- ◆HAProxy：用于负载均衡
- ◆Redis：用作分布式缓存层
- ◆CruiseControl.NET：用于代码构建和自动化部署
- ◆Lucene.NET：用于搜索
- ◆Bacula：用于备份
- ◆Nagios：（n2rrd 和 drraw 插件）用于监控
- ◆Splunk：用于日志
- ◆SQL Monitor：Red Gate 公司提供，用于 SQL Server 监控
- ◆Bind：用于 DNS
- ◆Rovio：一个小巧的机器人（真正的机器人），让远程开发人员可以通过“虚拟方式”访问办公室。

- ◆Pingdom：外部监控和警报服务网站

开发人员和系统管理员

- ◆14 名开发人员
- ◆2 名系统管理员

架构的更多信息和汲取的经验

◆使用了 Proxy，而不是使用 Windows 网络负载均衡（NLB），因为 HAProxy 成本低廉、易于使用，还是免费的；而且通过 Hyper-V，很适合作为网络上的一个 512M 虚拟机“设备”。它还在服务器的前端工作，所以对服务器来说完全透明；而且作为不同的网络层，更容易排除故障，而不是与你的所有窗口配置混杂在一起。

◆没有使用 CDN，因为与捆绑在现有主机方案中的带宽相比，连亚马逊 CDN 这样“便宜的”CDN 其费用都非常昂贵。按照亚马逊的 CDN 费率和 Stack Overflow 的带宽使用量每月至少要付 1000 美元。

◆备份到磁盘上，便于快速恢复；备份到磁带上，便于历史归档。

◆SQL Server 的全文搜索机制集成度非常差，问题多多，功能很弱，所以 Stack Overflow 改用了 Lucene。

◆最受关注的是峰值 HTTP 请求数字，因为这正是他们需要确保能处理的方面。

◆所有产品如今都在同一个 Stack Exchange 平台上运行。那意味着 Stack Overflow、Super User、Server Fault、Meta、WebApps 和 Meta Web Apps 都在同一个软件上运行。

◆有一些独立的 StackExchange 站点，因为人们拥有不同的专业技能，这些技能并不适用于不同的主题站点。你也许是世界上最出色的大厨，但并不是说你就有能力修复服务器。

◆Stack Overflow 尽量把一切都放到缓存中。

◆匿名用户访问的所有页面通过输出缓存（Output Caching）放到缓存中，随后提供给匿名用户。

◆每个站点有三种不同的缓存：本地缓存、站点缓存和全局缓存。

◆本地缓存：只能通过 1 对服务器/站点来访问。

◆为了限制网络延迟时间，Stack Overflow 使用了本地“一级”缓存（HttpRuntime.Cache），缓存服务器上最近设定/读取的值。这样就可以把网络上的缓存查找开销减小至 0 字节。

◆缓存里面含有用户会话和等待的视图数更新等内容。

◆缓存完全驻留在内存中，没有网络或数据库访问。

◆站点缓存：可以由一个站点（任何服务器上）的任何实例来访问。

◆大部分缓存的值进入到这里，热点问题 ID 列表和用户验收率就是两个典例。

◆缓存驻留在 Redis（位于不同的数据库，纯粹为了易于调试）。

◆Redis 的速度很快，缓存查找中速度最慢的部分就是读取字节并写到网络上。

◆值被发送到 Redis 之前先进行压缩。Stack Overflow 有许多处理器，大部分数据是字符串，所以得到的压缩比很高。

◆Redis 机器上的处理器使用率为 0%。

◆全局缓存：全局缓存被所有站点和服务共享。

◆缓存内容包括收件箱、API 使用限额和另外几项真正全局的内容。

◆缓存驻留在 Redis 中（位于数据库 0，同样为了易于调试）。

◆缓存中的大部分项目在超时（通常是几分钟）后过期，从来不被明确删除。需要宣布某个特定的缓存项目无效时，他们使用 Redis 消息传递机制，向“一级”缓存发布删除通知。

◆知名软件工程师、Fog Creek Software 公司首席执行官 Joel Spolsky 不是微软的忠诚分子，他并不为 Stack Overflow 做出技术决策，认为微软的许可证是个舍入误差。

◆Stack Overflow 为自己的输入/输出系统选择了英特尔 X25 固态硬盘组成的 RAID

10 阵列。这个 RAID 阵列消除了可靠性方面的任何问题；与 FusionIO 相比，固态硬盘的性能确实很好，而价格又便宜得多。

◆微软许可证的总标价约为 24.2 万美元。由于 Stack Overflow 使用 Bizspark，所以没在支付总标价，但他们能付的最多也就这么多。

◆Intel 网卡取代了 Broadcom 网卡和主生产服务器。这解决了他们之前面临的问题：连接中断、数据包丢失和地址解析协议（ARP）表损坏。■※

原文：

<http://highscalability.com/blog/2011/3/3/stack-overflow-architecture-update-now-at-95-million-page-vi.html>

<http://os.51cto.com/art/201103/248135.htm>

推荐阅读：

[专访人人网黄晶：SNS 网站后台架构探秘](#)

[揭秘企业级 web 负载均衡完美架构（图）](#)

[大型 B2C 网站高性能可伸缩架构技术探秘](#)

[国内外大型 SNS 网站后台架构对比](#)

在 2011 年 2 月到 2011 年 3 月间，发生了下面这些事.....

八卦，趣闻与数字 2011.2 – 2011.3

整理/51CTO 系统频道

【UNIX】UNIX 的销售在 2010 年最后一个季度保持平稳，而 LINUX 系统的销售额则猛增了几乎 30%。同期 Windows 服务器系统的销售额增长了 17%。

<http://os.51cto.com/art/201103/247991.htm>

【Debian】根据 Distrowatch 统计，在 323 款目前仍然活跃的 Linux 发行版中，有 128 款基于 Debian，74 款基于 Ubuntu。换句话说，所有 Linux 发行版中有 63% 是源于 Debian。

<http://os.51cto.com/art/201103/247743.htm>

【RHEL】红帽近日已改变了发行企业版 Linux 内核代码的方式，此举旨在防止甲骨文和 Novell 抢走其客户，让这些竞争对手更难弄清楚在哪些地方打上了补丁。

<http://os.51cto.com/art/201103/247842.htm>

【Solaris】McNealy：“如果我们只是牢牢抓住英特尔奔腾芯片，然后结合 Solaris 制作单项或者双向的匹萨盒，那么 Linux 可能永远都不会出现了，今天的 Google 也将会在 Solaris 上运行。”

<http://os.51cto.com/art/201103/246527.htm>

【Fedora】代号为 Lovelock 的 Fedora 15 的 alpha 版本已经发布可以测试了，目前 Beta 版的代码开发也已完成，正式版计划在 5 月份发布。

<http://os.51cto.com/art/201103/248116.htm>

【Ubuntu】Ubuntu 11.04 Natty 之后的 11.10 版开发代号已确定，代号名为 Oneiric Ocelot，其含义为“梦幻般的虎猫”。

<http://os.51cto.com/art/201103/247858.htm>

【LibreOffice】Ubuntu 官方开发成员已经表示 LibreOffice 3.3 将成为 Ubuntu 11.04 的默认的办公套件。

<http://os.51cto.com/art/201102/246429.htm>

【SUSE】Novell 宣布已经为大量使用 SAP 软件的用户客户化一种版本的 SUSE Linux 企业服务器软件。这个软件的名称是“SUSE Linux Enterprise Server for SAP Applications”

<http://os.51cto.com/art/201102/245778.htm>

本期专题：备份



有言道，冗余不做，日子甬过；备份不做，十恶不赦。

本专题收录了备份常用工具的介绍与一些生产环境的备份操作案例。

dump 执行类似 tar 的功能。然而，dump 倾向于考虑文件系统而不是个别的文件。

Linux 下的开源数据备份和恢复软件

文/李洋

(1) Amanda

Amanda 是由马里兰大学的 James da Silva 在 1991 年所开发的。它是一个复杂的网络备份系统，能够把 LAN 中的所有计算机备份到一台服务器的磁带驱动器、磁盘或光盘上。Amanda 本身并不是备份程序，它其实只是管理其他备份软件的封装软件。它使用系统上的 dump 和 restore 命令作为底层的备份软件，同时也能够使用 tar 命令，针对于 windows 计算机，Amanda 还能够使用 smbtar 命令来实现备份。

Amanda 支持类型广泛的磁带驱动器，并且能够使用磁带驱动器中的硬件压缩功能，或者也可以在数据通过网络之前使用客户机的 compress 以及 gzip 命令来压缩备份。其次，

Amanda 能够使用临时保存磁盘作为备份存档的中间存储媒介，以优化磁带的写入性能并保证在磁带出错时也能备份数据。

下载地址：

<http://www.amanda.org>

(2) BackupPC

BackupPC 是一个高效的企业级别的备份系统，可以备份桌面系统，笔记本等。支持 Linux 和 Windows 操作系统的备份，目前常用于 Linux 操作系统。其最新的版本为 3.1.0。它主要具有如下几个显著特点：

- ◆最小化使用服务器的磁盘空间和 I/O 来对数据进行备份和恢复；
- ◆采用优化的压缩技术，进一步减少磁盘空间需求量；

◆具有强大和友好的图形用户界面，方便用户操作使用；

◆支持多种备份方式和备份选项；

◆灵活的配置方法。

下载地址：

<http://backuppc.sourceforge.net/>

(3) Bacula

bacula 采用模块化设计，采用 c/s 构架，理论上可以把任意多台主机的资料备份到任意多台，而用户不需要在每台机器上都写一个配置文件控制他们运作，所有主要的工作都在一台主控备份机器上控制。登陆上主控备份用户就可以清楚地知道和监控什么备份正在运行什么备份成功了，什么备份失败了，所有的备份日志也会集中到用户指定的地方，这样的集中式管理机制让管理工作更简单一点。恢复的时候也很简单，简单运行几个命令用户就可以把指定的备份恢复。支持完全备份、差分备份增量备份等多种备份机制；支持把备份写到多种备份媒质中，比如写入到硬盘文件中，也支持写到磁带中。支持平台相当多，设置包括

win 平台 (备份 win , 还不支持备份到 win)。下载地址 :

<http://www.bacula.org/en/>

(4) Xtar

Xtar 是 Linux 系统桌面环境下查看和处理 tar 的工具。tar 工具是 Unix 备份文件的工具 , Linux 继承了该工具。因为 tar 几乎可以工作于任何环境中 , 所以 Linux 老用户都信赖它。但是 tar 是一个命令行的工具 , 没有图形界面 , 使用起来不是非常方便。tar 命令的参数非常多 , 常用的包括 30 多个 , 初学者往往不易掌握。因此 , Xtar 是一个图形化的 tar 工具。Xtar 全部兼容 tar 命令 , 可以在桌面环境下完成打包整个目录树的任务 , 这使得它特别适合用于备份。Xtar 与文件系统无关 , 可以使用在 ext2 、 ext3 、 JFS 、 Reiser 和其它文件系统上 , 支持各种备份介质 : 软盘、光盘、可重写的光盘 JazZip、磁带等。

(5) Taper

Taper 是一款拥有良好用户界面的磁带备份和恢复软件 , 并且是开放源代码的。它可以从一台磁带上备份 / 恢复软件 , 支持自动更新备份和恢复 , 是一个相当好用的工具。Taper 可以运行在命令行和 KDE、Gonme 桌面环境下。在命令行下启动一个终端输入即可启动。Taper 最大的优点是支持多种备份介质磁带、软盘、ZIP 驱动器、硬盘等。毕竟磁带的价格还是比较昂贵的。下载地址 :

<http://taper.sourceforge.net/>

(6) Arkeia

Arkeia 是一个基于客户 / 服务模式的备份解决方案。Arkeia 使用了独特的多线程技术 , 因此备份速度很快。如果客户端是选择用图形界面的 , 需要 Java 虚拟机 (JVM) 支持 (或者 JDK)。作为商业软件 Arkeia 和 Kdat 相比最大的优点是备份速度快 (在高速网络中 Arkeia 一个小时可以备份 700GB 文件)、和广泛应用的平台 (通过 Samba 可以备份 Windows 主机)。如果使用图形界面需要安装两个软件 : arkeiasb-server 、

arkeiasb-gui。Arkeia 有非常详细的在线帮助 , 很容易上手。下载地址 :

<http://www.arkeia.com/download/>

(7) webCDcreator

目前政府和企事业单位改换为 Linux 用于桌面系统。针对这种情况 , 如若需要实现备份可以在安装刻录机的 Linux 服务器部署 webCDcreator 共享刻录软件。然后网络中的所有节点 (Linux、Windows 都可) 都可使用这台刻录机进行数据备份。首先在服务器端安装配置软件 , 然后在客户端通过安装 Java 的浏览器即可访问服务器上的刻录机。部署 webCDcreator 刻录软件的优点 : 节约资金 : 所有软件都是免费的。集中化的方式有利于管理。下载地址 :

<http://joerghaeger.de/webCDcreator/>

(8) Ghost for Linux

Ghost for Linux 一个出色的硬盘对拷工具 , 它可以最大限度地减少用户每次安装操作系统的时间。它能适用于不同大小的硬盘 , 而且空间都被充分利用起来。注意 Ghost for

Linux 不是赛门铁克的 Ghost，它是一个开放源代码软件，支持 ATA、serial-ATA 和 SCSI 硬盘。使用 Ghost for Linux 进行 Linux 系统备份的优点在于：无需购买磁带机设备，使用一个大硬盘就可以为多台系统进行全盘备份（一台磁带机的价格相当于 100 个硬盘的价格）。可以使用匿名传输的方式把备份文件传输到 FTP 服务器。缺点在于：备份时间比磁带机长。下载地址：

<https://sourceforge.net/projects/g4l/>

(9) NeroLinux

随着 Linux 内核版本的增加，Linux 操作系统越来越大，用 DVD 刻录机备份操作系统成为必然。NeroLinux 是 Nero 公司发布的一款桌面环境下的 CD/DVD 刻录软件，在操作上 NeroLinux 与 Windows 的版本基本相同，它支持 2.4/2.6 版本内核，并且将采用 RPM 和 Debian 包等易安装软件包形式。从备份介质方面 NeroLinux 支持所有主流刻录盘片。它有非常容易使用的界面，功能比较齐全。下载地址：

<http://www.nero.com>

(10) mkCDrec

MkCDrec（建立可恢复系统的只读 CD）是 Linux 系统中建立可引导灾难恢复 CD 的工具。它支持多数文件系统、RAID（廉价冗余磁盘阵列）以及可选的工具，可以用来进行系统故障修复。使用 mkCDrec Cd 进行引导系统，用户还需做其它工作：如果用户选择建立系统备份，那么用户可以执行

</etc/recovery/start-restore.sh>

该命令运行后将提示用户完整的系统恢复过程。如果您在建立启动 CD 时没有进行备份，那么系统会提示没有数据可以进行恢复。/etc/recovery 目录包含了许多工具可以用来进行系统恢复或者修复崩溃的服务器，在 mkCDrec 的 Web 站点上有详细的说明。下载地址：

<http://mkcdrec.sourceforge.net/>

(11) tar 和 cpio 命令

tar 是一个已移植到 Linux 中的经典 Unix 命令。tar 是 tape archive（磁带归档）的缩写，最初设计用于将文件打包到磁带上，现在我们大都使用它来实现备份某个分区或者某

些重要的文件目录。我们使用 tar 可以打包整个目录树，这使得它特别适合于备份。归档文件可以全部还原，或从中展开单独的文件和目录。备份可以保存到基于文件的设备或磁带设备上。文件可以在还原时重定向，以便将它们重新放到一个与最初保存它们的目录（或系统）不同的目录（或系统）。tar 是与文件系统无关的，因而它的适用范围很广，它可以使用在 ext2、ext3、jfs、Reiser 和其他文件系统上。

使用 tar 非常类似于使用诸如 Windows 环境下的 Winzip、WinRAR 这样的文件实用工具。只需将它指向一个目的（可以是文件或设备），然后指定想要打包的文件，就可以通过标准的压缩类型来动态压缩归档文件，或指定一个自己选择的外部压缩程序。要通过 bzip2 压缩或解压缩文件，可使用 tar -z 命令。

cpio 命令可以从 tar 或者 cpio 档案文件中拷入或者拷出文件。cpio 命令和 tar 命令兼容，但是这个命令具备一些 tar 命令没有的功能，如下所示：

- ◆支持 cpio 和 tar 两种档案文件格式；

- ◆支持许多老式磁带数据格式；
- ◆能够通过一个管道 (pipe-line) 读取文件的文件名。

目前看来，只有很少的 Linux 软件包是以 cpio 格式发行的。用户如果对 cpio 命令的详细情况感兴趣，可以通过 “man cpio” 命令阅读它的使用手册。

(12) dump 和 restore 命令

dump 执行类似 tar 的功能。然而，dump 倾向于考虑文件系统而不是个别的文件。dump 检查 ext2 文件系统上的文件，并确定哪些文件需要备份。这些文件将出于安全保护而被复制到给定的磁盘、磁带或其他存储媒体上。在大多数媒体上，容量是通过一直写入直至返回一个 end-of-media 标记来确定的。配合 dump 的程序是 restore，它用于从转储映像还原文件。restore 命令执行转储的逆向功能。可以首先还原文件系统的完全备份，而后续的增量备份可以在已还原的完全备份之上覆盖。可以从完全或部分备份中还原单独的文件或者目录树。dump 和 restore 都能在网络

运行，因此用户可以通过远程设备进行备份或还原。dump 和 restore 使用磁带驱动器和提供广泛选项的文件设备。然而，两者都仅限用于 ext2 和 ext3 文件系统。如果使用的是 JFS、Reiser 或者其他文件系统，将需要其他的实用工具

dump 默认已经安装在 Linux 中。使用 dump 命令时，它会建立一份自上次备份操作以来进行修改过的文件列表，然后把这些文件打包成一个单独的文件。在做备份时，需要指定一个备份级别，它是 0-9 之间的一个整数。级别为 m 的转储会对从上次进行的级别小于 m 的转储操作以来修改过的所有文件进行备份，而级别 0 就是完全备份。通过这种方式，可以很轻松的实现增量备份、差分备份，甚至每日备份。通常情况下，第一次备份时可选择级别 0，以后每天做增量备份时就可以每天依次使用级别 1，级别 2，级别 3 等等，这样随着级别的增加以后每次做增量备份时才会对以前的各个老版本进行及时地更新；当需要每天做差异备份时，与上述原则也一样。

Dump 备份具有如下几个显著的优点：

- ◆备份可以跨多卷磁带；
- ◆能备份任何类型的文件，甚至是设备；
- ◆备份时，文件的权限、用户、用户组、修改时间等都会被一一准确保存；
- ◆能够正确处理从未包含任何数据的文件块
- ◆能够做增量、差分备份。

然而，在使用过程中，需要牢记该备份方法的一些缺点，以便在使用中驾轻就熟，扬长避短：

- ◆每个文件系统必须单独转储备份；
- ◆只有本地计算机的文件系统才能够转储备份；
- ◆由于 dump 可以将备份存储在磁带上。Linux 通常用 /dev/st0 代表倒带设备，而用 /dev/nst0 代表非倒带设备，使用倒带设备存储时，当磁带用完它会自动倒带并接着存储，所以会覆盖以前的数据，这样就存在以前数据丢失的风险，这点尤其需要注意。■※

原文：

<http://netsecurity.51cto.com/art/201003/190437.htm>

这样的话在备份的过程中，数据不能像 SQL 读取的时候保持事务上一致。只有当备份的数据文件在数据库里复原了，他们才又一致了，这类似于数据库 down 掉之重启一样。

Facebook 是怎么做 MySQL 备份的?

文/Nagavamsi Ponnekanti
编译/Tim Chou

Facebook 的用户每天创造大量的数据，为了确保数据可靠的存储，我们每天进行数据备份。我们通过将原来的逻辑备份改成定制化的物理备份，显著地提升了备份的速度(不增加体积的情况下)。

从 mysqldump 到 xtrabackup

我们使用 mysqldump 来进行每日的数据库备份，mysqldump 对数据进行逻辑备份，就像应用访问数据库那样，mysqldump 以 SQL 语句的方式从数据库中读取一张张表，将表结构 and 数据转保存到文本文件。mysqldump 最大的问题是速度太慢(对于我们的一些大的数据库，通常要花 24 小时，甚至更久)，并且以 SQL 语句的方式读取数据可能造成磁盘的

随机读，这就会造成主机的 load 增大，影响性能。对于时间太长，我们可以跑多个实例来并发的做备份，这能缩短备份的时间，但是会造成更多的 load，更影响主机的性能。

另外一个可行的备份方式是进行物理备份(我们称之为二进制备份，binary backup)，通过操作系统层面，读取数据库磁盘文件，而非通过 SQL 语句。这样的话在备份的过程中，数据不能像 SQL 读取的时候保持事务上一致。只有当备份的数据文件在数据库里复原了，他们才又一致了，这类似于数据库 down 掉之重启一样。

我们通过修改增强 xtrabackup 来满足我们额外的需求：

1. 支持快速的表级还原
2. 增强全量和增量备份
3. 支持混合增量备份

xtrabackup 支持增量备份，也就是备份自上次全量备份后改变的数据。这样我们就能够减少备份的空间(比如每天一次增量备份，每周一次全量备份)。xtrabackup 也支持多级增量备份，不过我们不使用，避免复杂。

1. 表级还原

我们写了一个 PHP 脚本，来从二进制备份文件中读取并还原指定的表。当前，这个脚本还不能自己从备份文件中读取信息创建表结构因此必须事先准备好一个对应的空的表。我们对 xtrabackup 也做了相应的修改来支持这个工具。这个修改就是支持 xtrabackup 导入导出单表。单表的还原比全量还原快得多，因为只需要从文件中读取相应的表的信息。

2. 调整全量和增量复制

fb 是 xtrabackup 早期的增量备份功能的用户，起初对于一些有大量表的数据库，

xtrabackup 的增量备份不起作用。后来我们和 percona 一起解决了这些问题。

xtrabackup 只有本地增量备份功能，也就是说增量备份的文件必须要和 mysql 在同台主机上。我们修改使之支持远程增量备份，也就是通过类似管道的方式将备份的数据同时发送到远程主机。如果先在本地做增量备份，然后通过网络传到远程主机，对我们来说是不可取的，因为会大大增加本地的写操作。

xtrabackup 以 1MB 为 1 个 chunk 来读取数据库文件，我们发现使用 8MB 时，能够使增量备份的速度快一倍，使全量备份快 40%。

3. 让增量备份成为真正的增量

xtrabackup 的增量备份读取数据库的每个 page，来判断哪些 page 改变了。我们创建了一个 page 追踪器，通过读取事务日志，并且通过每个表的 bitmap，来追踪修改过的 page。这样我们就能很好的追踪哪些 page 改变了哪些没变，我们就可以只读取那些改变过的 page。我们称这位真正的增量备份。

不过讽刺的是，我们发现这种真正的增量备份比普通的增量备份反而来的慢。这是因为普通的增量备份用 8MB 的 chunk 来读取文件而真正的增量备份读取文件的大小是不定的，从 16KB (INNODB 中一个 page 的大小) 到 8MB，这取决于有多少连续的 page 是被修改过的。因此在我们的很多场景下 (自上次全量备份后大概 10%-30% 的 page 修改了)，真正的增量备份比普通的增量备份花了更多的 IO 调用。

因为我们进行了改进，有了一种混合增量的备份，通过避免读取未修改的 page 来减少 IO 次数，在我们的场景下，这种混合增量备份减少了 20%-30% 的 IO，IO 的大小从 16KB 到 8MB 不等。

下表描述了使用上述改进的方法来处理大概 750GB 数据时产生的不同结果。由于 mysqldump 速度的原因，mysqldump 只在少数几个数据库上跑，我们使用 gzip 来对 mysqldump 的结果进行压缩，很慢，但是压缩率很高。

QPress 用来压缩二进制备份，它比 gzip 快多了，但是压缩效率更低。由于我们经常作增量备份，较少作全量备份，整个二进制备份所需要的空间和 mysqldump 所需要的空间还是差不多的。■ ※

	Time	Size after Compression
mysqldump	~30 hours	~76GB
Full backup	~3 hours	~235GB
Incremental after 1 day	2 hrs 22 minutes	~38GB
Hybrid incremental after 1 day	1 hr 54 minutes	~38GB
Incremental after 2 days	2 hrs 23 minutes	~50GB
Hybrid incremental after 2 days	1 hr 58 minutes	~50GB

图 各种备份方式速度和大小比较

原文：

[Hybrid Incremental MySQL Backups](#)

译文：

<http://class-too.us/?p=530>

推荐阅读：

[MySQLdump 增量备份、完全备份与恢复](#)

[MySQL mysqldump 命令的正确应用](#)

[MySQL 数据库备份与恢复](#)

[透过差异谈完全备份、增量备份与差异备份](#)

管理员的最高境界就是花力气去懒惰，我很向往这个境界。所以，我花了些时间写了个 shell，就让系统自己去干这些事情吧！

Linux 中常用的 Shell 备份脚本三则

收集整理/51CTO 系统频道

有言道，冗余不做，日子甬过；备份不做，十恶不赦。对于 Linux 系统管理员而言，备份是要定期做的重要工作。虽然工作性质重要但这并不妨碍系统管理员通过各种方式让系统自动完成这个任务。下面提供的脚本是 Linux 系统上常用的备份脚本，供大家参考。

Linux 常用备份脚本

文/波大帅哥

备份网站内容

```
#!/bin/bash
#指定运行的脚本 shell
#运行脚本要给用户执行权限
bakdir=/backup
month=`date +%m`
day=`date +%d`
year=`date +%Y`
```

```
hour=`date +%k`
min=`date +%M`
dirname=$year-$month-$day-$hour-$min
mkdir $bakdir/$dirname
mkdir $bakdir/$dirname/conf
mkdir $bakdir/$dirname/web
mkdir $bakdir/$dirname/db
#备份 conf，检测通过
gzupload=upload.tgz
cp /opt/apache2/conf/httpd.conf
  $bakdir/$dirname/conf/httpd.conf
cd /opt/apache2/htdocs/php
tar -zcvf $bakdir/$dirname/web/
  $gzupload ./upload
#远程拷贝的目录要有可写权限
scp -r /backup/$dirname
  root@10.1.1.178:/backup
```

备份数据库：

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

```
#指定运行的脚本 shell
#运行脚本要给用户执行权限
bakdir=/backup
month=`date +%m`
day=`date +%d`
year=`date +%Y`
hour=`date +%k`
min=`date +%M`
dirname=$year-$month-$day-$hour-$min
mkdir $bakdir/$dirname
mkdir $bakdir/$dirname/conf
mkdir $bakdir/$dirname/web
mkdir $bakdir/$dirname/db
#热备份数据库
cp /opt/mysql/my.cnf $bakdir/
  $dirname/db/my.cnf
cd /opt/mysql
mysqldump --opt -u zhy -p
  --password=1986 test>$bakdir/
  $dirname/db/test.sql
mysqldump --opt -u zhy -p
  --password=1986 phpwind>$bakdir/
  $dirname/db/phpwind.sql
#远程拷贝的目录要有可写权限
scp -r /backup/$dirname
  root@10.1.1.178:/backup
```

原文：

<http://bobo520.blog.51cto.com/514552/116798>

MySQL 的热备份脚本

文/抚琴煮酒

本脚本是 mysqldump --opt 的补充：

```
#!/bin/bash
PATH=/usr/local/sbin:/usr/bin:/bin

# The Directory of Backup
BACKDIR=/usr/mysql_backup
# The Password of MySQL
ROOTPASS=password

# Remake the Directory of Backup
rm -rf $BACKDIR
mkdir -p $BACKDIR

# Get the Name of Database
DBLIST=`ls -p /var/lib/mysql | grep /
| tr -d /`

# Backup with Database
for dbname in $DBLIST
do
mysqlhotcopy $dbname -u root -p
$ROOTPASS $BACKDIR | logger -t
mysqlhotcopy
done
```

原文：

<http://os.51cto.com/art/201010/229129.htm>

svn dump 全自动备份脚本

文/carywu

subversion 作为开发部门的一个必备工具，其地位已经从原来的小弟一跃成为老大的位置（真有取代 CVS 的可能）。平时我都是很细心的呵护着我公司的这台服务器，冷不丁打个喷嚏，我都会颤抖三分。

svn 服务器代码库是每时每刻都在更新的，所以，最好的备份方式就是同步了，正好 svn 新版本带了 svnsync 的命令，可以实现这个功能。再结合 post-commit 的 hook 脚本，我们就可以在用户提交代码的同时，让服务器自动更新库版本到备份服务器上。

最近要给美国的 SVN 服务器做升级，由于这台服务器有一定年份了，在本公司也算是个三朝元老，保存了多少 code 啊！为了保险起见，老板将迁移的工作郑重的交到我的手上。在欣欣然的同时也颤抖着。

不过还好我已经对其轻车熟路，在新的服务器上安装好 apache 和 subversion 软件后，开始着手备份与迁移了。

由于旧服务器版本太低，没有 svnsync 的功能，考虑再三，我选择使用了 svnadmin dump 去做备份。

可是有一个问题了，每个库的版本都有几万了，然后服务器上有十几个库，我可没耐心一个个输入 svnadmin dump 去备份它们，那会让我觉得长夜漫漫，难于睡眠！

管理员的最高境界就是花力气去懒惰，所以，我花了些时间写了个 shell，就让系统自己去干这些事情吧！

Shell 功能：

只需要大家修改源库所在的目录，和备份存放的目录就可以了。

第一次运行会全部备份，然后以后每次运行会增量备份。

所以，恢复的时候，先恢复全部备份，然后依次按照日期恢复增量备份即可。

（完整的备份脚本见下页）

```
#!/bin/sh

#####
# this script is used for svn repository backup
# written: carywu
# date: 2008/11/21
#####

#####
# mail address for status updates
# - This is used to email you a status report
#####
MAILADDR=carywu@yahoo.cn

#####
# backup directory, please change it
# - This is the path to the backup directory
BACKUPDIR=/svnroot/svnback

#####
# svn repository directory, please change it.
# - This is the path to the directory you want to archive
#####
SVNDIR=/svnroot/repository

#####
# HOSTNAME and project list
# - This is also used for reporting
#####
ProjectLst=$BACKUPDIR/projectlist.txt
```

```
#####
# svn backup log path and date
# - This is used to log result
#####
LogFile=$BACKUPDIR/svnback.log
DATE=`date +%Y%m%d-%T`
#####
# svn command path
# - This is the path to the directory you want to archive
#####
export PATH=$PATH:/bin:/usr/bin:/usr/local/bin

# our actual rsyncing function
do_accounting()
{
    echo " " >> $LogFile
    echo " " >> $LogFile
    echo "#####" >> $LogFile
    echo "$DATE" >> $LogFile
    echo "#####" >> $LogFile
    cd $BackDir
}

do_svndump()
{
    PROJECTLIST=`cat $ProjectLst`
    cd $SVNDIR
    for project in $PROJECTLIST
    do
```

```
echo "begin to dump $project databases" >> $LogFile
if [ ! -f $BACKUPDIR/$project.dump ]
then
YOUNGEST=`svnlook youngest $project`
svnadmin dump $project > $BACKUPDIR/$project.dump
echo "OK,dump file successfully!!"
echo "$YOUNGEST" > $BACKUPDIR/$project.youngest
else
    echo "$project.dump existed,will do increament
job" >> $LogFile
    if [ ! -f $BACKUPDIR/$project.youngest ]
    then
        echo "error, no youngest check!" >> $LogFile
    else
        PREVYOUNGEST=`cat $BACKUPDIR/
$project.youngest`
        NEWYOUNGEST=`svnlook youngest $project`
        if [ $PREVYOUNGEST -eq $NEWYOUNGEST ]
        then
            echo " no database updated!" >> $LogFile
        else
            LASTYOUNGEST=`expr $PREVYOUNGEST + 1`
            echo "last youngest is $LASTYOUNGEST" >>
$LogFile
            svnadmin dump $project --revision
$LASTYOUNGEST:$NEWYOUNGEST --incremental > $BACKUPDIR/
$project-$LASTYOUNGEST-$NEWYOUNGEST.$DATE
            echo "$NEWYOUNGEST" > $BACKUPDIR/
$project.youngest
        fi
    fi
fi
```

```
fi
done
}

# our post rsync accounting function
do_mail()
{
    mail $MAILADDR -s svn-back_log < $LogFile
}
# some error handling and/or run our backup and accounting
do_accounting && do_svndump && do_mail
```

原文：

<http://canywu.blog.51cto.com/13185/207571>

推荐阅读：

[SVN 同步备份面面观及俺的脚本](#)

[SVN 自助更新：运维利器 Puppet 实例讲解（一）](#)

[在 Eclipse 里使用 SVN 必装 Subclipse 插件](#)

[CentOS 系统中安装 subversion 并使用 svn+ssh 访问](#)

[SVN 版本控制中文使用指南：《Tortoise Subversion 使用手册》](#)

[64 位 FreeBSD 8.1 下搭建 SVN 服务器](#)

[四步介绍如何用 GoogleCode 和 SVN 来进行 CodeReview 和版本控制](#)

随着网站的运行 MySQL 数据库也越来越大，备份问题也日益凸显，使用 phpmyadmin 导出或导入时可能会因为超时文件过大等原因无法导入或导出。

Linux VPS/服务器上导入导出 MySQL 数据库

文/VP Ser

随着网站的运行 MySQL 数据库也越来越大，备份问题也日益凸显，使用 phpmyadmin 导出或导入时可能会因为超时文件过大等原因无法导入或导出。下面提供几个解决方法：

MySQL 数据库导出

方法 1: mysqldump 命令

执行命令：

```
/usr/local/mysql/bin/mysqldump -u 用户名 -p 数据库名 > 文件名
```

如果数据库用户名有密码密码，执行后会提示输入密码。如果数据库用户名没有密码，就将命令中的“-p”参数删除。

注意：推荐使用 MySQL root 帐号进行操作，因为有些数据库可能设置了只能对应的用户进行操作。

方法 2: phpmyadmin

phpmyadmin 支持将数据库文件直接备份到配置文件指定的目录，默认安装的 lnmp 使用了修改后的配置，可以将数据库直接备份到 /home/wwwroot/phpmyadmin/save/ 下面然后用户通过 http 或者 ftp 等方式下载即可。

方法 3：第三方软件

如：帝国备份王。专门为 MySQL 大数据的备份与导入而设计的稳定高效软件，系统采用

分卷备份与导入，理论上可备份任何大小的数据库。[点击查阅帝国备份王技术文档。](#)

MySQL 数据库导入

方法 1：mysql 命令

执行命令：

```
/usr/local/mysql/bin/mysql -u 用户名 -p 数据库名 < 文件名
```

参数与 mysqldump 的使用一样。

注意：导入的数据库名需要已经存在。

方法 2：phpmyadmin

phpmyadmin 可以将数据库文件导入到指定的数据库里。可以上传数据库备份文件到 /home/wwwroot/phpmyadmin/upload/ 目录进行导入。

方法 3：第三方软件

如：帝国备份王。✖

本文为节选，原文见：

<http://www.vpser.net/manage/mysql-import-export.html>

其实，大部分管理员已经有数台 Linux 服务器，所以考虑能否安装类似 Time Capsule 的服务让 Linux 为 Mac 提供网络备份。

自建基于 Linux 的 Time Capsule 服务

文/明城

Apple 的 Time Machine 提供了非常完备的数据备份方案，但每次使用外接磁盘备份很麻烦也很容易忘记备份。同时，个人对于 Time Capsule “高昂的” 售价（虽然它的确值那些钱）颇有微词。

其实，大部分管理员已经有数台 Linux 服务器，所以考虑能否安装类似 Time Capsule 的服务让 Linux 为 Mac 提供网络备份。

如果您和我一样有类似的需求，那么这篇文章可能就是你想要的：^D

安装软件包

本人使用的 Linux 发行版为 Debian，其他发行版的也应该类似。

首先，安装需要的软件包：

```
% sudo apt-get install netatalk
% sudo apt-get install avahi
# 通常这个包已经安装了的
% sudo apt-get install nss-mdns
```

注意：如果需要自己编译安装 netatalk 包，则需要加入 ssl 支持，具体看编译脚本参数。

配置

上面的软件包正常安装完毕以后，接下来就是配置了。

服务器端

1、编辑 /etc/default/netatalk，找到对应的内容，更改为

```
ATALKD_RUN=yes
PAPD_RUN=no
CNID_METAD_RUN=yes
AFPD_RUN=yes
TIMELORD_RUN=no
A2BOOT_RUN=no
```

2、编辑 /etc/netatalk/afpd.conf，在最底部加入行：

```
- -tcp -noddp -uamlist
uams_dhx.so,uams_dhx2.so
-nosavepassword
```

3、编辑 /etc/nsswitch.conf，在最底部加入行：

```
hosts: files mdns4_minimal
[NOTFOUND=return] dns mdns4 mdns
```

4、新增

/etc/avahi/services/afpd.servic

内容为：

```
<?xml version="1.0" standalone='no'?>
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
  <name replace-wildcards="yes">
    %h</name>
    <service>
      <type>_afpovertcp._tcp</type>
      <port>548</port>
```

```
</service>
<service>
  <type>_device-info._tcp</type>
  <port>0</port>
  <txt-record>model=Xserve</txt-record>
</service>
</service-group>
```

这个时候 netatalk 协议已经配置完毕，我们要共享出个目录让 Mac 映射为磁盘。编辑

```
/etc/netatalk/AppleVolumes.default
```

加入下面的行：

```
~/TimeMachine "TimeMachine"
allow:mingcheng cnidscheme:cdb
options:usedots,upriv,tm
```

简单说明下：

其中的 mingcheng 为用户名，它指定了哪些用户应用该规则。如果有多个用户，则使用“,”逗号分割。

~/TimeMachine 为用户主目录下的 TimeMachine 目录，也就是我们要保存备份数据的目录。

"TimeMachine" 为映射磁盘的名称，便于辨别起见，通常不用更改。

较关键的地方是 "options:tm"，这个选项其实就是让 netatalk 充当 Time Machine 服务器的意思 (via)。

Tips：由于备份数据比较大，可以考虑 mount --bind 映射到 Linux 本地的其他合适的分区或目录 (详细)。

至此，基本的 Linux 服务器端配置完毕，重启服务 (重启需要花点时间)

```
% service netatalk restart
% service avahi-daemon restart
```

重启服务以后，在 Mac 端连接 Linux 服务器。



图1 连接服务器

输入对应的用户名和密码以后，即可挂载已经配置好的磁盘。

Mac 端

在 Mac 端，我们还需要做些配置。打开终端 (Terminal)，执行下面的命令使 Time Machine 能够使用网络磁盘作为备份盘：

```
defaults write
com.apple.systempreferences
TMShowUnsupportedNetworkVolumes 1
```

由于部分兼容性问题，还需要另外手工制作磁盘疏散镜像。打开“偏好设置”中的“共享”，可以看见“电脑名称”，例如我的是“明城的 Macbook Pro” (注意大小写和空格)，这里将其命名为 <computer_name>



图2 设置电脑名称

然后，打开终端粘贴如下面的命令获得本机的网卡 Mac 地址，这里将其命名为 <mac address>


```
ifconfig en0 | grep ether | awk  
'{print $2}' | sed 's://g'
```

如图，例如我的是

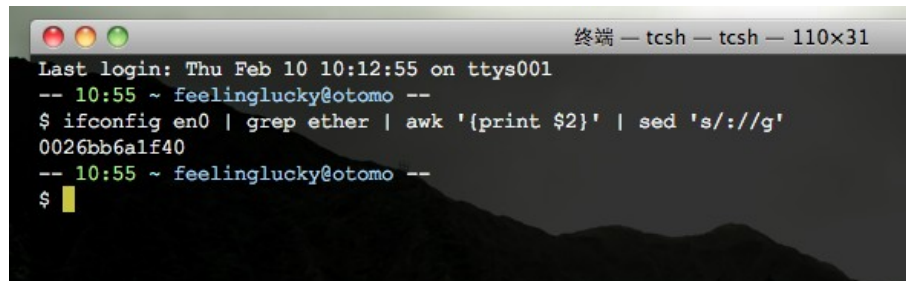


图 3 获取 mac 地址

然后运行下面的命令，注意将上步骤获得的 <computer_name>、<mac address> 代入命令行：

```
sudo hdiutil create -size 320g -type  
SPARSEBUNDLE -nospotlight -volname  
"Backup of <computer_name>" \  
-fs "Case-sensitive Journaled HFS+"  
-verbose  
~/Desktop/<computer_name>_<mac  
address>.sparsebundle
```

注意：-size 320g 参数是本机的硬盘容量，请根据实际情况更改。

最终在桌面会有个类似我本机的文件名（注意空格等字符）：

明城的 Macbook
Pro_0026bb6a1f40.sparsebundle

然后将这文件扔到上面配置好的 Linux 网络磁盘根目录。打开 TimeMachine 选择对应的磁盘，即可以开始备份。

Tips：首次网络备份会比较缓慢，建议插入网线。

一般问题

问：为什么我选择网络磁盘以后，TimeMachine 一直在“正在准备磁盘”？

答：这是磁盘疏散镜像没有制作正确，或者你命名的文件名不正确。可以暂停备份，挂载网络磁盘看 TimeMachine 为你自己生成的文件名，然后在将上面步骤生成的文件替换。

问：TimeMachine 的备份数据是加密过的吗？

答：抱歉，保存在服务器上的数据是没有加密的（数据传输如有 SSL 是加密的）。
※

原文：

<http://www.gracecode.com/archives/3057/>

通过 dd 命令实现 MBR 备份恢复

文/张勤

MBR (Master Boot Record, 主引导记录) 是计算机开机后访问硬盘时所必须要读取的首个扇区。MBR 记录着硬盘本身的相关信息以及硬盘各个分区的大小及位置信息，是数据信息的重要入口。如果 MBR 受到破坏，硬盘上的基本数据结构信息将会丢失，需要用繁琐的方式试探性的重建数据结构信息后才可能重新访问原先的数据。MBR 内的信息是 fdisk 格式化的时候写入的。

在 Linux 下使用如下命令备份 MBR：

```
[root@srv ~]# dd if=/dev/sda1  
of=/root/mbr.bak bs=512 count=1
```

同样也可以使用 dd 命令恢复 MBR：

```
root@srv ~]# dd if=/root/mbr.bak  
of=/dev/sda1 bs=512 count=1
```

推荐阅读：

[开机流程与主引导分区 \(MBR\)](#)

[ghost FreeBSD 恢复后修复 MBR](#)

网卡也支持另外一种被称为混杂的模式，可以接收网络上所有的帧，比如说很多用物抓包工具就需要让网卡运行在这个模式下。

Debian 下通过 Bonding 实现双网卡单 IP

文/张勤

Linux 双网卡绑定一个 IP 地址，实质工作就是使用两块网卡虚拟为一块，使用同一个 IP 地址，是我们能够得到更好的更快的服务。其实这项技术在 Sun 和 Cisco 中早已存在，被称为 Trunking 和 Etherchannel 技术，在 Linux 的 2.4.x 的内核中也采用这这种技术，被称为 bonding。本文以 Debian 为例介绍这个技术的实现。

bonding 的原理

在正常情况下，网卡只接收目的硬件地址 (MAC Address) 是自身 Mac 的以太网帧，对于别的数据帧都过滤掉，以减轻驱动程序负担。但是网卡也支持另外一种被称为混杂的模式，可以接收网络上所有的帧，比如说很多用

物抓包工具就需要让网卡运行在这个模式下。bonding 也运行在这个模式下，而且修改了驱动程序中的 MAC 地址，将两块网卡的 MAC 地址改成相同，可以接收特定 MAC 的数据帧。然后把相应的数据帧传送给 bond 驱动程序处理。

bonding 配置

使用如下命令安装 bonding。

```
srv10:~# apt-get install ifenslave
```

使用如下命令让系统开机自动加载模块 bonding。其中 miimon 参数是用于进行链路监测的。比如：miimon=100，那么系统每 100ms 监测一次链路连接状态，如果有一条线路不通就转入另一条线路；mode 的值表示 bonding 工作模式。

```
srv10:~# sh -c "echo bonding mode=6  
miimon=100 >> /etc/modules"
```

修改 /etc/network/interfaces 文件为如下内容。

```
auto lo  
iface lo inet loopback  
auto bond0  
iface bond0 inet static#虚拟网卡的  
TCP/IP 配置  
address 192.168.159.100  
netmask 255.255.255.0  
gateway 192.168.159.1  
dns-nameservers 192.168.159.1  
post-up ifenslave bond0 eth0 eth1  
pre-down ifenslave -d bond0 eth0 eth1
```

如果在安装 ifenslave 后没有重新启动计算机，必须手动加载 bonding 模块。

```
srv10:~# modprobe bonding mode=6  
miimon=100
```

使用如下命令重新启动网卡。

```
srv10:~# /etc/init.d/networking  
restart
```

本文为节选，原文见：

<http://os.51cto.com/art/201102/245483.htm>

rsync 以 delete 操作时要注意，它会完全同步这边的目录与服务器的目录中的文件，这样会产生一个后果，它会删掉服务器中多余的文件，这样做很危险。如果是线上机器，请慎之又慎，切记！

CentOS 5.5 下 rsync 使用技巧与权限问题解读

文/抚琴煮酒

Rsync 使用所谓的“Rsync 算法”来使本地和远程两个主机之间的文件达到同步，这个算法只传送两个文件的不同部分，而不是每次都整份传送，因此速度相当快。

rsync 的优点

- ①可以镜像保存整个目录树和文件系统；
- ②可以很容易做到保持原来文件的权限、时间、软硬链接等；
- ③无须特殊权限即可安装；
- ④优化的流程，文件传输效率高；
- ⑤可以使用 rsh、ssh 等方式来传输文件，当然也可以通过直接的 socket 连接；
- ⑥支持匿名传输

服务器端的配置

具体的就不多说了，重要的地方说一下。首先检查 rsync 是否安装：

```
rpm -q rsync
rsync-2.6.8-3.1
```

说明 rsync 已安装，如果提示出现 package rsync is not installed 则说明这个软件没有安装大家可以使用 yum 进行安装

另外，关闭防火墙和 SELinux，因为是内网中传输，所以这些没必要

```
service iptables stop && chkconfig
iptables off
setenforce 0
```

下面分享下我自己定义的配置文件/etc/rsyncd.conf。文件我先贴出来，方便线上环境的朋友直接复制，后面再写上详细注释。

```
uid = nobody
gid = nobody
user chroot = no
max connections = 200
timeout = 600
pid file = /var/run/rsyncd.pid
lock file = /var/run/rsyncd.lock
log file = /var/log/rsyncd.log
[backup]
path=/backup/
ignore errors
read only = no
list = no
hosts allow =
    192.168.21.0/255.255.255.0
auth users = test
secrets file = /etc/rsyncd.password
```

注释：

```
uid = nobody
```

进行备份的用户，nobody 为任何用户

```
gid = nobody
```

进行备份的组，nobody 为任意组

```
use chroot = no
```


如果 "use chroot" 指定为 true , 那么 rsync 在传输文件以前首先 chroot 到 path 参数所指定的目录下。这样做的原因是实现额外的安全防护, 但是缺点是需要以 root 权限, 并且不能备份指向外部的符号连接所指向的目录文件。默认情况下 chroot 值为 true. 但是这个一般不需要, 我选择 no 或 false

```
list = no
```

不允许列清单

```
max connections = 200
```

最大连接数

```
timeout = 600
```

覆盖客户指定的 IP 超时时间, 也就是说 rsync 服务器不会永远等待一个崩溃的客户端。

```
pidfile = /var/run/rsyncd.pid
```

pid 文件的存放位置

```
lock file = /var/run/rsync.lock
```

锁文件的存放位置

```
log file = /var/log/rsyncd.log
```

日志文件的存放位置

```
[backup]
```

这里是认证模块名, 即跟 samba 语法一样, 是对外公布的名字

```
path = /backup/
```

这里是参与同步的目录

```
ignore errors
```

可以忽略一些无关的 IO 错误

```
read only = no
```

允许可读可写

```
list = no
```

不允许列清单

```
hosts allow =  
192.168.21.0/255.255.255.0
```

这里跟 samba 的语法是一样的, 只允许 192.168.21.0/24 的网段进行同步, 拒绝其它一切

```
auth users = test
```

认证的用户名

```
secrets file = /etc/rsyncd.password
```

密码文件存放地址

注意: 对于这个地方的配置文件, 虽然解释的很清楚, 可是一些朋友始终还会在这里出错 这里我提亮点易错点:

1、[backup] 认证模块名和 path = /backup/ 参与同步的目录

这里的 path 大家要记好了, 这里不要随便的一设置就直接完事, 要知道这里是认证模块的, 以后从客户机备份的数据会存储在这里。

2、auth users = redhat 认证的用户名

这个名字是服务器端实实在在存在用户, 大家不要直接跟步骤走却忽略了这点。如果服务器端少了这个的话我估计你的数据同步就实现不了, 大家要谨记。

3、path = /backup/ 参与同步的目录

这个需要稍后自己要在根目录下自己建

```
cd /  
mkdir backup  
chmod -R 777 /backup  
echo "test:test" > /etc/rsyncd.password
```

(这里我设置的是用户名和密码一致)

为了安全起见, 我设置他的权限为 600

```
chmod 600 /etc/rsync.password
```

客户端配置:

```
echo "test" > /etc/rsyncd.password
```

这里仅仅只需要密码, 不需要用户了, 免得要同步时还要手动互动

```
chmod 600 /etc/rsync.password
```

配置完了之后，可能会遇到一些权限问题：

问题一：

```
@ERROR: chroot failed
rsync error: error starting client-
server protocol (code 5) at
main.c(1522) [receiver=3.0.3]
```

原因：

服务器端的目录不存在或无权限。

创建目录并修正权限可解决问题。

问题二：

```
@ERROR: auth failed on module tee
rsync error: error starting client-
server protocol (code 5) at
main.c(1522) [receiver=3.0.3]
```

原因：

服务器端该模块（tee）需要验证用户名密码，但客户端没有提供正确的用户名密码，认证失败。

提供正确的用户名密码解决此问题。

问题三：

```
@ERROR: Unknown module 'tee_nonexists'
rsync error: error starting client-
server protocol (code 5) at
main.c(1522) [receiver=3.0.3]
```

原因：

服务器不存在指定模块。

提供正确的模块名或在服务器端修改成你要的模块以解决问题。

测试

都配置完毕之后就是测试了。在 FreeBSD 机器上，输入命令

```
rsync -vzrtp --delete
/home/andrewy/etc
test@192.168.21.41::backup --password-
file=/etc/rsyncd.password
```

查看同步效果。

注意：我这里是直接以 root 进行操作。rsync 以 delete 操作时要注意，它会完全同步这边的目录与服务器的目录中的文件，这样会产生一个后果，它会删掉服务器中多余的文件。如果是线上机器，请慎之又慎，切记！

原文：

<http://os.51cto.com/art/201101/243374.htm>

相关阅读：

[Linux 远程备份工具 Rsync 使用案例](#)

Down 友交流区

以下为针对[第五期 down 友部分反馈](#)的回复。

闪电 CTO：不知道多长时间更新一期？

答：《趋势》发布时间为每个月的第二个星期五。

hanyedeleng：Stallman 最新访谈在网站上有啦。

答：《趋势》的内容都是收录自互联网，在征得作者许可后刊登。相同的，投递到《趋势》的稿件也均会先在 51CTO 系统频道发布。

changjun977：希望加点 Ubuntu 的知识：)

答：Ubuntu 有专门的粉丝杂志 Full Circle（有中文版），可以关注下。选题合适的情况下，《趋势》也会考虑做一期以 Ubuntu 为中心的专题。

jackyduys：能否介绍下 linux 下双机的操作？

答：已将该主题列入备考选题：)

mayc82：如果内容再多一些就好了。

答：目前《趋势》的篇幅控制在了 30 页以内，是为了大家能够比较轻松的阅读。如果您认为内容多些比较好，可以到[《趋势》专题页面](#)参与投票。

招募启事

《Linux 运维趋势》的建设需要您的加入！

您可以通过如下方式参与我们杂志的建设：

1、推荐文章

无论是您在互联网上看到的好文章，还是您自己总结/整理的资料；无论是英文还是中文；无论是入门的还是高端的，都欢迎推荐！推荐方式包括：

a) 在技术圈中分享：<http://g.51cto.com/linuxops>

b) 发邮件给编辑：yangsai@51cto.com

2、投稿

如果您认为自己在 Linux 方面具有专家级别的能力，并且有与大家分享您技术经验的热诚，同时也有兴趣挣点稿费花花，那么欢迎您的投稿！

如果您在 IT 技术方面的翻译有很高的能力，能够快速、高质量的完成译文，并且也经常浏览到一些 Linux 方面的优秀外文，那么也欢迎您的投稿，或加入我们 51CTO 的翻译团队！

投稿邮箱：yangsai@51cto.com

3、推广与意见

如果您喜欢我们的杂志，认为这本杂志对于您的工作有所帮助，请向您的 Linux 好友、同事们推荐它！

如果您觉得这份杂志还有什么地方需要改进或补充，也希望您能够提出您的宝贵意见！

联系人：yangsai@51cto.com

下期预告

下期主题为：迁移。欢迎关注！

本刊为月刊，预定每月发布日期为：

每个月的第二个星期五

您可以通过如下方式检查新刊发布：

1、通过电子邮件订阅：

订阅方式请参考本杂志的专题页：

<http://os.51cto.com/art/201011/233915.htm>

2、经常光顾 51CTO Linux 频道：

<http://os.51cto.com/linux/>

《Linux 运维趋势》是由 51CTO 系统频道策划、针对 Linux/Unix 系统运维人员的一份电子杂志，内容从基础的技巧心得、实际操作案例到中、高端的运维技术趋势与理念等均有覆盖。

《Linux 运维趋势》是开放的非盈利性电子杂志，其中所有内容均收集整理自国内外互联网（包含 51CTO 系统频道本身的内容）。对于来自国内的内容，编辑都会事先征求原作者的许可（八卦，趣闻&数字栏目例外）。如果您认为本杂志的内容侵犯到了您的版权，可发信至 yangsai@51cto.com 进行投诉。