

开源时代

OPEN SOURCE TIMES

2009年4月刊 总第八期



ORACLE

ORACLE TO BUY SUN

开源业界:

Ubuntu计划在中国大陆成立公司
Sun发布MySQL5.4 响应速度提升90%?

社区扫描:

Ubuntu9.04发布 新增上网本专用版
Moblin项目易主 Linux基金会全面接管

Oracle74亿美元收购Sun 开源业界如何面对?

本期推荐: Oracle74亿美元收购Sun, 重量级事件。Sun将何去何从, 特别等开源项目, 新东家Oracle将对这些网友和业界专家将为您分析这些项目

应该是过去一个月中最震惊IT业界的是Sun控制的Java、MySQL和OpenOffice“开源资产”如何处理? 来自CU的资深的未来命运。



行业观察:

甲骨文收购Sun
或整合OpenSolaris和Linux
开源软件真正用起来在于执行力

技术新知:

p2p VPN的今生前世
Linux下SVN+apache简单安装
与配置

技术沙龙:

北京负载均衡技术沙龙
上海网络优化高端论坛

网友来信:

Linux为何难以推广
——高中生手记





2009年3月刊 总第七期

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

甲骨文74亿美元收购Sun 细节将逐渐公布	005
OpenSUSE Build Service将加入LDN	005
联想: Linux太复杂很难让普通消费者接受	006
Sun发布MySQL5.4 响应速度提升90%?	007
IBM利用EnterpriseDB技术 蚕食甲骨文客户	008
Ubuntu 9.04没有获得Oracle认证	008
Ubuntu计划在中国大陆成立公司	008
第三届Linux基金会合作峰会召开 将发力移动设备市场	009
Linux基金会: Linux是增长速度最快的平台	010
中科红旗Linux7桌面正式版月底发布	011
红帽瞄准虚拟化市场暂无意Linux桌面	011
叫好不叫座 OpenMoko开源手机停止研发	013
Symbian移动开发平台开放源码	013
SUSE Linux Enterprise 11 北京发布会 主打虚拟化	014

社区扫描

Ubuntu 9.04发布 新增上网本专用版	016
Moblin项目易主 Linux基金会全面接管	016
MySQL创始人宣布要创建MySQL分支	017
Intel投身社区 资助GCC编译器开发	018
顶级脚本语言: PHP、Ruby、Python之争	019
传谷歌拟削减对火狐支持 挖角Mozilla	021
开源社区争论话题之二: 走向歧路的标准组织	022
微软资助密歇根州立大学开发NFS v4 windows版本	023
Red Hat公布“开源世界地图”	024

行业观察

甲骨文收购Sun 或整合OpenSolaris和Linux	025
开源风投：回报时刻即将到来	026
IDC：2009年Linux软件销售收入将增21%	026
让开源软件真正用起来在于执行力	027
开源软件开发成本高达3870亿美元	028
微软TomTom和解案留下许多问号	028
又一开放精神与既得利益的拉扯战争	029
Oracle需要的是Zend而不是Red Hat	030
惠普称正在评估Android 尚未敲定是否采用	031
手机Linux安全之路 如何免于步履蹒跚	031

技术沙龙

沙龙服务社区 交流走向成熟	033
---------------	-----

专家专栏

MySQL数据库技能 互联网行业从业必备	038
----------------------	-----

网友来信

我们的开源世界需要更 Easy	040
-----------------	-----

本期推荐

042

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，我们将会在第一时间做出处理。

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。

技术新知

MFS分布式文件系统架设笔记	047
P2P VPN的今生前世	051
CPU负载平衡之一--运行队列的load计算	055
Linux下SVN+apache简单安装与配置	064
DirectFB 之一：PC平台编译与安装	076
OpenBSD. Nginx. MySQL. PHP环境搭建手册 (v2)	083

网友热评

100

卷首语

自去年3月以来, ChinaUnix 已经组织了9次技术沙龙, 例如去年6月的网络流量分析沙龙、去年9月的负载均衡沙龙, 再到最近的运维监控沙龙, 以及网络优化论坛——上海站, 我们的技术沙龙得到了大家的热情支持, 在此表示深深的感谢。

在已经组织的系列沙龙中, 主要集中在北京, 这是因为我们总部在北京, 便于组织和管理。然而, 这种在单一城市组织的现象并不好。事实上, 不管是在论坛讨论时, 还是在 MSN 的交流中, 很多人都希望我们能够在全国范围内组织沙龙活动。应该说这种希望就是一种动力, 只有大家的积极支持, ChinaUnix 技术沙龙的全国化布局才能得到展开。4月11日在上海进行的网络优化论坛, 是我们 ChinaUnix 技术沙龙全国化布局的重要一步; 5月我们将移师广州; 紧接着, 6月我们将计划举办西安站、成都站。下一阶段, 我们还将走入天津、青岛、武汉、郑州、沈阳、杭州和南京。我们的技术沙龙全国化布局已经全面展开, 为大家提供良好的技术交流平台, 真诚地希望大家更加热情地支持我们!

另外, 今年9月, ChinaUnix 将主办 2009 Linux 技术大会, 按设想, 这将是国内 Linux 业界规模空前的盛会。之所以想主办这次大会, 同当前的 Linux 的发展密切相关。

近几年来, Linux 开源技术在国内得到了蓬勃的发展, 越来越多的企业开始部署 Linux 系统平台, 将企业的关键业务系统部署在 Linux 系统平台, 例如, 众多的网站类企业, 将企业的关键业务部分建在了 Linux 平台上。最近两个月来, 有好几个负责运维的网友跟我说, 近半年时间里, 他们单位已将一半的 Windows Server2003 的服务器换成 Linux 平台了。这种现象正在越来越多的企业里发生, 例如制造、物流企业等。就种趋势也说明了 Linux 的未来是美好的。

同样, 前几天我看了一份报告, 说当前的 Linux 平台职位需求已经接近 Windows 平台职位的需求, 比 2008 年又有了新的进步, 基于 Linux 的运维和开发人员非常受欢迎。在经济危机的今年, Linux 更加显示出其威力。

正是在这种背景下, 我们准备主办 2009Linux 技术大会, 在这次大会里, 我们计划邀请 Ebay、facebook 等国际知名企业的一线技术主管参会, 另外, 还将邀请国内顶级技术企业的技术总监、架构师进行主题演讲。我们相信这将有效地推动国内 Linux 和开源技术的应用和发展。

只有大家的大力支持, Linux 就会发展得更好!

唐川

开源业界

甲骨文 74 亿美元收购 Sun 细节将逐渐公布

北京时间 4 月 20 日晚消息，甲骨文宣布该公司将以每股 9.5 美元的价格收购 Sun。该交易总值将超过 70 亿美元。这一消息颇令人感到意外。本月早些时候，IBM 和 Sun 之间的收购谈判破裂。知情人士称，Sun 拒绝了 IBM 每股 9.4 美元的出价。



Sun 股价在盘前交易中上涨了 37.5%至 9.2 美元；甲骨文股价则下跌了 4.5%至 18.34 美元。Sun 向美国证券交易委员会提交的最新文件显示，其流通股约为 7.45 亿股。按每股 9.5 美元计算，Sun 的身价约为 71 亿美元。

针对甲骨文 74 亿美元收购 Sun 一事，Sun 大中华区新闻发言人 20 日晚间接接受采访时说，他已经于北京时间当日晚 7 点从内部获悉了这一收购案，Sun 大中华区对这一消息暂不置评，具体细节将逐步对外公布。

有关更多甲骨文收购 Sun 相关内容，请阅读本刊[本期推荐](#)栏目，有来自业内的多位专家和 CU 资深网友发表的观点和看法，特别是甲骨文收购 Sun 一事，对开源世界的影响。敬请阅读。

OpenSUSE Build Service 将加入 LDN

2009 年 4 月 10 日，北京——openSUSE 项目与 Linux 基金会近日联合宣布，openSUSE Build Service 将加入 Linux 开发人员网络（LDN）之中。openSUSE Build Service 是唯一可使开发人员针对所有主要的 Linux 发行版进行软件包装的开发平台，它可为创建完整的 openSUSE 发行版提供透明化的基础架构。另外，openSUSE 项目，一个由 Novell 赞助，并得到社区支持的开源项目，还宣布推出 openSUSE Build Service 新版本，该版本具有支持 ARM 平台编译的能力。

“我们的使命是不仅为 Linux 开发人员提供信息，而且还提供真正的工具，使他们能够在多种 Linux 发行版上尽可能方便地部署应用，此次所发布的内容是这个使命的一部分。” Linux 基金会市场推广与开发人员项目副总裁 Amanda McPherson 说。“对于那些想要针对所有主要的发行版快速方便地进行产品包装的开发人员和 ISV 们来说，openSUSE Build Service 的推出是历史性跨越的一大步。”

Linux 基金会将通过 Linux 开发人员网络的站点提供 openSUSE Build Service 的接口，这样，开发人员就可以通过 LDN 为所有主要的 Linux 发行版创建应用包。除了 openSUSE 和 SUSE Linux Enterprise，openSUSE Build Service 还可以帮助开发人员为 CentOS，Debian，Fedora，Mandriva，Red Hat Enterprise Linux 和 Ubuntu 创建应用包。在 LDN 中增加 openSUSE Build Service 的策略是对 LDN 上流行的 AppChecker 应用的锦上添花，开发人员能够用它创建 Linux 移

动式应用。对于 LDN 帮助开发人员推出这样的移动式应用这个总体目标而言，Build Service 是一个理想的工具。

openSUSE 项目新推出的 build service 1.6 版包括对 ARM 平台编译包的支持，它主要用于嵌入式设备。跨基础架构构建的支持意味着开发人员可为 openSUSE，Ubuntu，Debian，和 Fedora 创建 RPM 或 Debian 包。这项工作一直由 5e DataSoft GmbH 贡献，它作为 openSUSE 社群的一部分可以增加对基于 ARM 的嵌入式设备的支持。5e 提供基于 openSUSE 的解决方案。



“在所有的 Linux 发行版当中，我们认为 openSUSE 是我们解决方案的最佳平台。” 5e DataSoft 首席技术官 Martin Mohring 说。“由于 openSUSE Build Service 是 GPL 的，因此我们一直能够添加新的特性，并与 openSUSE 项目一起将这些特性集成到正式的 build service 之中。”

Open-Xchange 首席技术官 Jürgen Geck 说，“openSUSE Build Service 使我们能够同时为所有主要的 Linux 平台构建 Open-Xchange – 而且使这个过程极具效率并保证最终的产品具备广泛的兼容性。这项服务是免费的，其基础软件结构是遵从 GPL 的，因此不存在锁死的问题。”

最新版本的 build service 还包括对 openSUSE 设备、直播 CD、可安装的 USB 映像、Xen 映像和 VMware 映像的支持。开发人员现在可以利用 build service 创建自己定制的 openSUSE 发行版。

openSUSE 社群经理 Joe 'Zonker' Brockmeier 说，“这是 openSUSE 项目多年努力的一个高峰。openSUSE Build Service 始终努力成为一个加速 Linux 普遍采用的工具。看到 build service 成为 Linux 开发人员网络的一部分并受到更大社群的欢迎，这令人十分欣慰。”

联想：Linux 太复杂很难让普通消费者接受

4 月 22 日消息，联想全球竞争分析师 Matt Kohut 在接受采访时表示，上网本和笔记本电脑都采用 Linux 的情况永远不会出现。作为一个操作系统，开源软件永远不会拜托边缘化的状态。

Kohut 对 Tech.Blorge 网站说，上网本没有让 Linux 成为主流。不满的消费者不愿意使用这种便宜的操作系统，而选择了容易设置的专有的 Windows 操作系统。Kohut 说，有许多安装 Linux 的上网本会节省 50 或者 100 美元。但是，从行业的观点看，这种上网本的退货率很高，因为人们不知道用 Linux 操作系统怎么使用。

联想分析师指出，必须要自己编译代码和上载数据让许多人不愿意使用 Linux，因为普通人仅想拥有一台电脑。去网站下载源代码太困难了。Linux 应该达到这种程度：如果你要插入一些东西，Linux 就能装载驱动程序并且立即可以使用。

Kohut 说，这显然是 Linux 社区自身的错误。他说，他没有看到 Linux 社区或者相关公司认真地努力让 Linux 操作系统对用户更友好，以便在上网本方面取得成功。

Sun 发布 MySQL 5.4 响应速度提升 90%?

Sun 在第七届 MySQL 展会上发布了其最新版开源数据库 MySQL 5.4 的技术预览版本, MySQL 5.4 在性能和可伸缩性上进行了重大改进。MySQL 5.4 支持 InnoDB 存储引擎扩展至 16 路 x86 服务器和 64 路 CMT 服务器, 同时也优化了子查询和 JOIN 功能, 将对特定查询的响应速度提升了 90%, 这些性能和可伸缩性的提升非常明显, 而且不需使用额外应用程序或 SQL 代码。



Sun 软件架构和 MySQL 团队副总裁 Karen Tegan Padir 在大会的主题演讲时表示: “不需要对应用程序进行任何修改, MySQL 5.4 将显著提高它们的性能和可伸缩性, MySQL 5.4 也更加适用于扩展 SMP 系统上的部署。”

MySQL 5.4 新增功能和性能提升:

- 1、可伸缩性提升: 支持 InnoDB 存储引擎扩展至 16 路 x86 服务器和 64 路 CMT 服务器, 性能提升了一倍;
- 2、子查询优化: 提高了分析查询操作的性能, 相比之前版本, 子查询执行时间缩短;
- 3、新的查询运算法则: 利用主存加快多路连接尤其是 MySQL 集群的执行速度;
- 4、改进了存储过程: 增强了 SIGNAL/RESIGNAL 功能的错误管理能力, 应用程序可以更轻松地依赖商业逻辑的存储过程;
- 5、完善了 prepared statements: prepared statements 中新增了对输出参数的支持;
- 6、改善了信息数据库: 为存储过程提供了更多的元数据存取方式, 开发人员在使用 ODBC 和 JDBC 之类的连接器时可以获取更多的信息;
- 7、改进了对 DTrace 的支持: 提高了 Solaris 操作系统上的 MySQL 的诊断和故障排除能力。

支持平台:

MySQL 5.4 适用于多种硬件和软件平台, 包括: 红帽企业版 Linux (RHEL)、Novell 的 SuSE Enterprise Linux、微软的 Windows、Sun 的 Solaris 10、苹果的 Mac OS X、Free BSD、HP-UX、IBM AIX、IBM i5/OS 和其他 Linux 发行版本。

目前 MySQL 5.4 技术预览版本仅支持 64 位 Linux 和 Solaris 10 系统, 正式版本将在今年晚些时候发布。

IBM 利用 EnterpriseDB 技术 蚕食甲骨文客户

4月24日消息，甲骨文宣布收购 Sun 后，IBM 将利用小型数据库软件厂商 EnterpriseDB 的技术蚕食甲骨文的客户。

据国外媒体报道称，EnterpriseDB 发言人称，甲骨文应用软件可以不加修改地在该公司的



PostgresPlus 上运行。IBM 数据库业务副总裁阿温德·克里须那表示，现在，甲骨文的应用软件也可以在 DB2Release9.7 上运行了，“利用 EnterpriseDB 的移植工具，精通甲骨文技术的数据库管理员将能够使用 DB2 数据库。EnterpriseDB 的技术使甲骨文的 PL/SQL 查询和存储过程可以在 DB2 系统上运行。”

EnterpriseDB 首席执行官爱德华·博亚简表示，IBM 使用该公司的甲骨文兼容性技术是“对其技术的认可。”

Ubuntu 9.04 没有获得 Oracle 认证

Ubuntu9.04 Jaunty Jackalope 在 4月23日同时发布桌面版和服务器版，尽管它加入了许多新特性，但是它将缺乏一项关键项目——Oracle 认证。尽管 Ubuntu 创始人 Mark Shuttleworth 承认 Oracle 在 Linux 领域的重要性，但是他对此并不感到担忧。他指出，“Oracle 是企业适用性的试金石，所以 Oracle 认证对我们来说非常重要，它能为 Ubuntu 的可靠性和强度提供证明。”

Oracle（甲骨文）公司目前拥有自己的 Oracle Enterprise Linux（基于 Red Hat），并对他们运行在 Red Hat 和 Novell 版本 Linux 上的应用程序进行认证。Shuttleworth 指出，目前他还没看到哪个组织的 Linux 服务器上运行着大量的 Oracle 应用软件，他认为这些组织仅需要大量的数据库服务器。

Ubuntu 计划在中国大陆成立公司

北京时间 4月17日消息，Linux 操作系统 Ubuntu 的商业赞助商 canonical 公司一位高层透露，该公司正计划在中国大陆成立子公司，以加速进入这个三年后将成为全球最大电脑市场的国家。

canonical 公司全球 OEM 服务总监 Jon Melamut 近日透露，该公司正计划在中国大陆成立公司，以加速进军这个市场。Jon Melamut 预计，三年后中国将成为全球最大的电脑市场。Canonical 公司 2年前开始进入台湾市场。并在 3个月前已经在中国台湾成立了公司，办公室位于著名的 101 大厦，目前已经有 11 名工作人员。

canonical 公司中国大陆负责人于立强说，目前 canonical 在大陆有 3 名员工，主要提供商务和技术支持服务，计划在未来 6 个月增加更多技术人员。据介绍，将加强与中国大陆的 PPLive、QQ、迅雷等独立软件开发商(ISV)合作，让其软件能够运行在 Ubuntu 操作系统上。Canonical 公司赞助的 Ubuntu 不仅仅是一个操作系统，还集成了各种常用的应用软件，让用户安装 Ubuntu 后即可使用。

Jon Melamut 还透露，预计今年年底前，有几家国内的电脑厂商将会推出预装 Ubuntu 的电脑。目前，在中国市场，线下零售店能买到预装 Ubuntu 的电脑只有戴尔公司一家。

Canonical 及其运营维护的 Ubuntu 开源社区将继续提供免费下载、自由使用以及分享给其他使用者的服务，Canonical 的盈利模式是为商业用户和电脑厂商提供技术咨询，定制开发、培训等服务。

多年来用惯了微软 Windows 的用户，可能对 Ubuntu 比较陌生，它是目前全球最受欢迎的 Linux 操作系统，特别是个人使用的桌面端 Linux 系统，中国及全球的“乌邦图”粉丝众多。总部位于英国的 canonical 公司，是 Ubuntu 的商业赞助商，它运营着 Ubuntu 社区及一家提供 Ubuntu 技术咨询、定制开发、培训的商业公司。

第三届 Linux 基金会合作峰会召开 Linux 将发力移动设备市场

当地时间 4 月 8 日至 10 日，一年一度的 Linux 基金会合作峰会在美国加州旧金山举行。Linux 基金会执行总裁 Jim Zemlin 就像一个热情的传道士一样，面对听众一遍遍强调着 Linux 平台的好处和潜力，还特别介绍了 Linux 最近在移动设备上所做的努力。



在本次峰会上，Zemlin 谈及了 Linux 在多个领域的进步，包括超级计算和嵌入式系统。“在有关计算的任何一个方面，Linux 都是增长速度最快的平台。”他表示，“Linux 的增长速度是目前任何其它平台的两到三倍！”事实上，近期的市场数据的确对 Linux 有利。IDC 不久前发布报告称，调查显示当前的经济困难加快了 Linux 的普及，有超过六成的企业用户将有 Linux 服务器的需求。

Zemlin 还强调了 Linux 在桌面端的潜力，他认为应该用与过去不同的眼光来重新看待桌面操作系统，或者可以亲切地称之为桌面终端。因为构成桌面的范畴在不断变化，从笔记本电脑到手机，从浏览器到汽车，这些现在都可以归于桌面端，而 Linux 处理起来显然具有优势。“以前我们每年都会想今年的桌面市场会不会属于 Linux。但从今年开始，我们可以问自己这样的问题：什么才是桌面？”Zemlin 显得信心十足。他说 Linux 已经开始“对客户端进行重新定义，”而商业性的竞争对手比如 Windows 等无法做到这样。

云计算的火爆也为 Linux 提供了机会，而 Linux 平台也已经在这一领域取得领先。Zemlin 说，Google 之所以能够成为 Google，正因为它采用了 Linux，而不是在 Windows 上运行。2008 年底，红帽 CEO Whitehurst 也在接受访谈中表示：“云计算将全部使用 Linux。”。

Zemlin 指出，Linux 除了价格之外还有在很多方面具有优势。“Linux 当然是最经济实惠的平台，人们完全支付得起，而除了价格，我们还应该看到 Linux 的其他重要方面，”他说，Linux 不仅仅是一种免费的平台，它在软件开发和共享应用等方面也具有很大优势。Zemlin 还强调了 Linux 的安全

性，并以芝加哥商业交易所使用的 Linux 交易系统为例，来证实 Linux 平台的安全性，因为这一以 Linux 为基础的交易平台，年均交易量高达万亿美元。也许正因为 Linux 处于蓬勃的上升发展阶段，所以 Linux 仍然需要标准来约束，Zemlin 表示，Linux 标准基础（Linux Standard Base）的制定正在努力进行中。但标准必须是开放的，并防止垄断的既成事实的出现，他补充说。如果使用者担心专利问题，那么已经有一些组织机构和应用程序供应商可以对 Linux 及开源软件用户提供帮助，比如 Linux Defenders。

随后，来自业界巨头 Intel 的开源技术中心总监 Imad Sousou 就 Intel 基于 Linux 创建的 Moblin 移动平台与参会者进行了探讨，他介绍了目前正在测试阶段的 Moblin2 的改进计划。上周，Intel 将 Moblin 项目正式转移到 Linux 基金会的管辖之下，他认为，大企业当不了开源项目的项目管理者角色，短期也许可以，但长期肯定是行不通的。Moblin 的新功能将改善用户体验作为重点，并且打算走出不断“克隆”20 年的操作系统的开发习惯。Moblin2 的系统平台将采用“快速启动”功能，目标是将系统启动时间减少到只需 2 秒钟，目前的 Moblin 是 5 秒钟。在 UI 开发方面，Moblin 应用杂波处理技术使动画框架得到了飞跃。具有物理图形和 3D 功能，“在我们能够看到的任何操作系统上，这可能是最好的动画框架”，Sousou 这样说。另外，Moblin2 还计划加入一个称为 Conman 的连接管理器，对语音和 WiFi 等提供支持。

周四上午，技术专家们就 Linux 内核进行了专项讨论，并谈到了下一代 Linux 的趋势。“下一个 Linux 内核很快会发布，我们会看到下一代 Linux 的样子。”Linux 内核主要维护人 Andrew Morton 在接受采访时表示，“它已经运行了大约一年，每天都在更新。”目前的 Linux 最新是 2.6.29，这是两周之前公布的。

最后，已被 Linux 更新为默认文件系统的 ext4 文件系统也被提及。今年刚上任的 Linux 基金会 CTO Ted Ts'o 认为，ext4 是一个高性能的文件系统，特别能满足数据库用户对文件系统的的功能需求。总的来说，它允许更大的文件系统，具有更好的性能。他预计 ext4 文件系统将在本月发布的 Ubuntu9.04 和 Fedora11 中被默认采用。

Linux 基金会：Linux 是增长速度最快的平台

Linux 基金会执行董事 Jim Zemlin 称，Linux 应用的增长速度比其它平台快二至三倍。随着企业在这次经济衰退期间整合自己的技术基础设施，Linux 应用的增长速度会更快。一位 IDC 分析师也表示了同样的观点。这位分析师在最近的研究报告中称，Linux 将比它的竞争对手更好地承受经济衰退。

Linux 基金会执行董事 Jim Zemlin 本周三在旧金山举行的 Linux 协作峰会上发表演讲时讨论了 Linux 平台的机会和 Linux 团体能够合作克服挑战和缓解风险的一些方面的问题。他介绍了 Linux 高速增长的意义，强调了 Linux 在当前的经济环境中应用的潜力。他解释说，Linux 从计算的每一个方面来说都是增长速度最快的平台。Zemlin 指出，经济衰退正在迫使企业整合自己的技术基础设施。Linux 比它的竞争对手更适合用于更广泛的环境并且应用的成本较低，因此，Linux 的市场份额在提高。

IDC 分析师 Al Gillen 似乎支持 Zemlin 的观点。Gillen 最近发表的新的研究报告预测称，Linux 操作系统能够比其它平台更好地承受经济衰退的影响，将成为商业性操作系统的强大的竞争对手，并且将成为操作系统市场长期的赢家。

Zemlin 在讲话中还对 Linux 的竞争对手进行了抨击。他演示了微软最近做的一个广告。这个广告显示一位有成本意识的消费者购买一台 Windows 电脑，而不购买价格更高的漂亮的 Mac 电脑。他滑稽地说，这是微软第一次试图在价格上进行竞争。他认为在价格战中 Linux 最终将会战胜微软。

中科红旗 Linux7 桌面正式版月底发布

中科红旗 Linux 桌面版在 2009 年初发布了一款“牵牛版”，这个版本对应的版本号是 V7.0 RC1，但由于该版本还处于研发阶段，并且还有一些待解决的问题，直至近日，记者才得知，红旗 Linux7.0 桌面正式版将在 4 月底正式发布。

根据介绍，从 09 年起，针对 Linux 开源技术的发展特点，红旗 Linux 对个人版产品线做了重要调整，新系统更加注重用户的体验，兼容性也会更好，并尽可能把最新、最炫的 DD 呈现给关注开源技术的“红 Fan 家人”们，也希望获得更多爱好者对红旗 Linux 产品的关注、反馈和支持。

关于 Linux 桌面系统正式版的命名目前还是一个未知数，据消息透露，新的名字有可能从社区征集，也不排除直接内部决定的可能。除了新系统的命名，中科红旗人技术研发人员还在做最后的测试，希望这个新的 Linux 能够带给大家一个全新的体验。

Ubuntu Server 9.04 将兼容 Amazon 的云计算

据国外媒体报道，新版本的 Ubuntu Linux 将为客户带来云计算的能力。计划于 4 月 30 日发布的 Ubuntu Server 将可以在物理服务器之间移植基于 KVM 的虚拟机，该功能与 VMWare 使用 VMotion 进行虚拟机移植相似。

KVM 是 Kernel Virtual Machine（内核虚拟机）的缩写，它是以色列 Qumranet 出品的开源管理程序。去年 RedHat 公司收购了 Qumranet。Qumranet 的开发人员 AviKivity 将 KVM 带入了 Linux 内核开发的工作中，2007 年更新的 Linux 内核中开始支持 KVM 虚拟化技术。

另外，Canonical 将增加集群管理软件，通过它来管理一组 Ubuntu 服务器。当虚拟负载被添加到簇中，Ubuntu 新增的云管理器就会决定由哪一个服务器来处理这个负载，这样就不再需要其它的管理操作了。它会根据管理员预先向管理器输入的决策方法来平衡复杂。

云管理器暂时只是一个技术预览特性，在 Ubuntu Server 9.10 发布以前还需要进行稳定性测试，而不能作为产品。根据 Ubuntu 的命名惯例（在版本号中标明发布月份），9.10 版将会在十月份发布。

计划在 4 月底发布的 Ubuntu Server 9.04 将具有云端功能，它将可以在 Amazon EC2 环境下运行。Canonical 的企业部总监 Steve George 在谈到他们使用 AmazonEC2 平台的感想时说，Ubuntu 的开发人员从一年前开始运用云计算。开发小组希望 UbuntuServer 的用户可以将 Ubuntu 运行在一组 x86 的服务器上，并将它们建成一个公司内部的兼容 Amazon 的云系统。你将可以在云中探索，George 说，Ubuntu 将会加入 Eucalyptus 项目进行开源 API 的开发，该项目通过开源代码将模仿 Amazon 的专有 API。他说大家都知道 Amazon 的 API，而 Eucalyptus 项目在这一领域有一个强大的团队。

Canonical 没有承诺将所有的 Eucalyptus 的 API 纳入 UbuntuServer 中，不过预计将会包括足够的 API 来让用户创建类似 Amazon 的云计算功能。虽然这个说法还未经证实，Canonical 还是希望在云计算的热衷用户中推广 Ubuntu。

红帽瞄准虚拟化市场暂无意 Linux 桌面

在经济环境不确定的今天，虚拟化就像一棵救命稻草。这棵“稻草”不仅对于厂商来说意义非凡，对于 IT 预算面临紧缩困境的用户也同样充满诱惑力。

2009 年 2 月 23 日，在全球 Linux 市场中占据 80% 份额的开源操作系统领导厂商红帽宣布了该公司在 2009 年的虚拟化策略和路线图。红帽大中国区总裁胡伯林在会上表示，红帽在未来 3-18 个月里面将提供包括“红帽企业 Linux、红帽企业虚拟化 Hypervisor、红帽企业服务器虚拟化管理、红帽企业桌面虚拟化管理”在内的一系列新产品，为用户提供从服务器到桌面的全面虚拟化，主要针对那些需要大规模部署虚拟机的企业级用户。

红帽在其一份公开的白皮书中宣称，红帽企业虚拟化平台将“提供经济高效、易于管理的解决方案，同时决不牺牲性能可扩展性或安全性。”在 VMware 占据虚拟化市场领导地位、微软 Hyper-V 凭借强大品牌营销实力开拓虚拟化版图的今天，作为挑战者的红帽把焦点聚集到了现有产品的弱点和不足之上。



随着红帽虚拟化战略和线路图的公开这个开源巨人开始将自己近阶段的发展重点转移到了虚拟化上，最近的一系列行动也正积极地实践着这一发展目标。近期，红帽除了自身在虚拟化领域不断拓展外，还和思科公司携手合作，他们将共同合作提供一个与红帽企业 Linux 软件和思科整合计算系统相结合的突破性解决方案，此方案能帮助客户在一个虚拟化和高性能数据中心环境下作关键任务的应用部署。另外，红帽和思科将合作提供全球性联合销售和服务，并且为客户提供开发部署和启动工具。

另外，红帽软件公司还在最近发布了它的基于 Eclipse 的开发工具集的升级软件，以便让开发人员制作丰富的互动应用程序和 SOA 应用程序。红帽的 JBoss Developer Studio 2.0 综合版增加了对企业应用的支持，支持企业应用程序、门户网站和通过企业服务总线连接到数据源。这个工具集的第一个版本是在 18 个月前发布的，最初仅限于让开发人员在网络层的 Seam 框架的基础上制作具有丰富的互动功能的 Java 企业版应用程序。

与此同时，大家在今年的早些时候也应该了解到红帽已与微软低调宣布在虚拟化技术上展开合作。红帽将参与微软的“服务器虚拟化验证计划”，而微软则参加红帽的“虚拟化认证计划”，这意味着用户能够放心地在微软或红帽的操作系统上，实现虚拟的红帽或微软的操作系统。同时，用户将得到两家公司的技术支持。

虽然红帽目前集中将精力放在虚拟化市场的拓展上，但相信不少用户还是对红帽在桌面领域的发展抱着期待。尽管有关何时实现 Linux 桌面版的话题，我们已经听到无数次“今年”式的口号了。虽然 Linux 在用户数量方面有长足的增长，但依然没有达到如红帽子这类 Linux 厂商所期望的水平。红帽子的 CEO Jim Whitehurst 甚至对 Linux 桌面版表示了怀疑的看法，他认为如今的 Linux 依然不是主流操作系统。

看来红帽 CEO 的这一番言论，可能要让这些支持者难受一阵子了。Red Hat CEO--Jim Whitehurst 近日在 Open Source Business Conference (OSBC) 上对 CW 记者表示，他个人很质

疑桌面 Linux 的实际用处，他认为桌面 Linux 不能给企业用户带来经济效益。Whitehurst 说：首先，他不认为桌面 Linux 能为用户挣钱。而且只有很少的人会用桌面 Linux 做关键的事务，因为他们根本就打算为这个系统付费。

Jim Whitehurst 甚至称红帽子目前根本没有推出这样产品的计划。虽然 Linux 在服务器领域取得了成功，但用户的需求已经相比过去发生了很大的变化，任何现在就想把 Linux 推入桌面级应用的想法都将被证明是愚蠢的。

也许挑战 Windows 是错误的选择。除了那些横在眼前的现存壁垒之外，Linux 厂商还需要对未来 5-10 年内可能会出现的障碍进行预测。看起来红帽子对此心知肚明，希望他们未来能给桌面用户带来惊喜。

叫好不叫座 OpenMoko 开源手机停止研发

北京时间 2009 年 4 月 8 日，OpenMoko 宣布停止对新手机的研发工作。梦想和现实终究是有距离，这一消息是在瑞士的 OpenExpo 上被宣布的。这款采用 Linux 作为操作系统的手机将不再继续开发，与此相关的员工也将被裁员，OpenMoko 将把目光聚集在非智能手机上。FreeRunner 之前号称是从硬件到软件全部开源的手机，不捆绑运营商，无需谈判和支付授权费用，但由于失败的产品设计，效率低下的宣传和工厂产能，再加上时间的拖延和更加炫目的开放 Android 系统的出现，令公众对这款产品失去兴趣。

从市场上来看，尽管 OpenMoko Neo FreeRunner 这款开源性手机获得不少好评，但是鲜有人买，并且由于金融危机和激烈的市场竞争等原因，OpenMoko 的首席执行官 Sean Moss-Pultz 才在瑞士的会议上做出了停止此款开源手机研发这一艰难决定。但是考虑到终端用户和与其合作的手机制造商，这一决定还不能完全实施，但是研发进度会放慢，并将最终停止。

Symbian 移动开发平台开放源码

据报道，Symbian 基金会为了鼓励开发商在 symbian 的开放源码创新，它设计了一个建立在德州仪器 OMAP 技术之上的新的 Zoom OMAP34x-II 移动开发平台(MDP)。

Symbian 基金会与其生态系统称它正在 Symbian OS 和 S60、MOAP 软件资产基础上创造最行之有效的、开放的、完整的移动软件平台。Symbian Os 是一个开放的移动操作系统。德州仪器称能帮助客户解决问题并开发新的电子设备，使世界更聪明、健康、安全、环保并且有更多的乐趣。

德州仪器称 Zoom OMAP34x-II 移动开发平台(MDP)是一个围绕德州仪器 OMAP3430 处理器的功能全面的评价平台的。此移动开发平台道出了对开放源码社区和高级的操作系统(HLOS)开发商的基本需求。该平台建立在 OMAP 技术之上，使它成为一种统一了嵌入式应用的低功率需求与智能手机和 MIDs 的 multimedia-rich 预期。

新的 Zoom OMAP34x-II MDP 被认为是第一个 Symbian 基金会的参考平台。这是一种测试设备，有能力运行各种移动操作系统，它可以由那些希望致力于开放源码平台用来测试代码。开发商也可以利用 MDP 来熟悉任何新的和即将推出的 Symbian 平台的发布。

“TI 有着支持 Symbian OS 的悠久历史我们为我们与 Symbian 基金会的持续合作伙伴关系感到兴奋。” TI 无线业务副总裁兼总经理 Pierre Garnier 说。

Garnier 补充道，这些 OMAP 3 软件开发平台和 OMAP34x-II MDP 方便了 Symbian 开发商社

区利用 TI 的全面功能、实惠的硬件平台，激发创新并帮助开发商策划塑造未来移动消费体验的下一代技术，。Zoom OMAP34x-II MDP 利用 OMAP3430 处理器的功能来推进下一代应用的技术整合。

OMAP34x-II MDP 的主要特点包括 4.1” WVGA 电容的多接触屏幕和可容易连接的完整的 QWERTY 键盘；Wireless 802.11、蓝牙、创造大量连接机会的调频网络；MMC/ SD 卡、SIM 卡、HDMI 和 3G 模块扩展槽，提供使用额外存储器和特点的通路；还有外部调试板，使全部发展的控制触手可及的。

Symbian 基金会计划在接下来的两年用 Eclipse 公共许可证将移动平台移入开放源码。这将使平台代码可免费提供给所有人并因此使开发商更广泛的参与。作为平台发展的一部分，基金会正在培养一个开发商、硬件制造商和其他合作伙伴的社区。

SUSE Linux Enterprise 11 北京发布会 主打虚拟化

Linux 与 Windows 的对抗愈演愈烈，就算是在 Linux 服务器操作系统市场，也是竞争激烈。近日，Novell 在北京宣布推出了面向新一代数据中心及桌面系统开发的操作系统——SUSE Linux Enterprise 11。



Novell 东亚区总裁张先民博士在接受采访时指出，SUSE Linux Enterprise 11 是业界最强的互操作性平台，它经过 Novell 内部反复测试后成功推出，倾注了 Novell 对于开源事业的热情和合作伙伴、客户的需求的热心关注。

张先民表示，“Linux 的市场推广必须讲究一定的战略，要根据客户和市场需求，采用避实就虚、扬长避短的战略。比如在服务器、工作站等市场，Novell 具有很大优势，而在新兴的上网本市场，也是 Linux 的一个发力点。”

更体贴企业客户需求

张先民告诉记者，“SUSE Linux Enterprise 11 能够在整个企业范围内提供创新技术。其在兼容性、虚拟化、绿色节能等方面提供了诸多功能。”张先民指出，“目前的一个发展趋势就是 Unix 会逐渐被 Linux 发展趋势取代，操作系统以 Windows 和 Linux 为主要，这两个系统的兼容性变得很重要，这也是为什么两年前微软和 Novell 建立了战略合作。”

据张先民介绍，SUSE Linux Enterprise 11 能够在整个企业范围内提供创新技术。该平台对 SUSE Linux Enterprise Server 和 SUSE Linux Enterprise Desktop 进行了重大改进，还提供了两种新的扩展——SUSE Linux Enterprise Mono Extension，这是业界唯一能够帮助客户在 Linux 平台上完全运行 Microsoft .NET 应用的产品；SUSE Linux Enterprise High Availability Extension，一种可确保关键任务应用的正常运行时间，并降低高可用性拥有成本的集群产品。

“我们在提高兼容性方面下了很大功夫，SUSE Linux Enterprise 11 可运行于各种领先的硬件平台，包括 x86-32、x86-64、Itanium、IBM POWER 和 IBM System z。这些都是基于客户需求的。”面对客户越来越多的虚拟化应用需求，Novell 也作了很大努力，在 Open source 方面，Novell 拥有支持虚拟化的技术 Xen，Novell 是这个技术的主要贡献者，Xen 技术具有绝对的优势和性能。SUSE Linux Enterprise 11 还进行了优化，以能够运行于各类主要系统管理程序并实现准本地（near-native）性能，这些管理程序包括 VMware ESX、微软 Hyper-V 和 Xen。

张先民还指出，另外，很重要一点就是降低能耗，Novell 的 SUSE Linux Enterprise 11 符合政府提出的“节能减排”的号召。

紧抓上网本等商机

对于桌面版本的推广，张先民把很大希望寄托在当前火爆的上网本市场。他表示，“我们认为，SUSE Linux 将在上网本和家电下乡中取得显著成长，新版本将更惹用户喜欢。”

张先民认为，“微软最需要调整的是价格策略，而我们跟微软不一样，Novell 产品的价格不高，我们需要在产品结构和功能上作调整。”“家电下乡的受众相对于大中城市的手中说来说，对 IT 不是那么熟悉，所以在 UI（用户界面）方面是必要作相当大程度的简化，让他们觉得使用起来是很简单的，我们在这方面作了很大的努力，他们要求操作简单，而且待机时间长，这些是我们和 PC 厂商经过反复调研得出来的结果，我们根据他们的需求对我们的产品进行调整，我们非常有信心在上网本领域占有领先地位，在电脑下乡方面有一个显著的成长。”

据介绍，SUSE Linux Enterprise 作为从桌面到数据中心解决方案的领先者，得到了微软和 SAP 的支持，领先地位得到加强。作为完整解决方案的一部分，SUSE Linux Enterprise 11 将会得到包括思科、戴尔、富士通西门子计算机公司（Fujitsu Siemens Computers）、惠普、IBM、英特尔、微软、NEC、甲骨文和思爱普集团（SAP AG）在内的 Novell 全球战略合作伙伴的支持，以及全体解决方案、集成、硬件和软件合作伙伴的支持。

之前与 Novell 有很好合作关系的联想、微星、惠普、戴尔、宝德等，都纷纷表示将加大与 Novell 的合作关系，近期将推出预装 SUSE Linux Enterprise 11 的服务器或 PC 产品。张先民透露，Novell 还积极与 OEM 合作伙伴联想、微星、惠普等一起涉足上网本市场。

社区扫描

Ubuntu 9.04 发布 新增上网本专用版

Canonical 终于官方发布了 Ubuntu 9.04 的正式版，除了惯有的桌面版和服务版之外还首次加入了上网本专用版 “Netbook Remix” (UNR)。



Ubuntu 9.04 桌面版基于 Linux Kernel 2.6.28，增加了大量提升用户体验的新功能、新特性：系统启动速度明显加快，最短只需 25 秒钟；增强的挂起恢复功能，可有效延长电池续航时间，并能让系统在休眠恢复后立即投入使用；Wi-Fi 和 3G 环境之间的智能切换支持更多无线设备和 3G 卡，能带来更平滑的无线体验。

Ubuntu 9.04 集成了桌面环境 GNOME 2.26 和新版办公套装 OpenOffice.org 3.0，后者完全兼容微软 Office，但却是免费的，不需要支付 200 美元。

新系统还第一次引入了一种新的整合式通知系统样式，将不同程序的通知方法融合在一起，整体更加简洁，也不会干扰用户体验。此外还有新的图标和壁纸等，ext4 文件系统也已实现，只不过默认仍是 ext3。

Ubuntu 9.04 服务器版整合了来自开源社区的最新稳定版程序，拓展了硬件兼容性，新功能则有支持 KVM 虚拟化、支持 Samba 文件服务器集群、更简单的邮件服务器设置和 Dovecot-Postfix 迁移等。

新系统还带来了新的“Ubuntu 企业云服务” (UEC)，并成为第一个能让企业在防火墙内部建立云环境的 Linux 发行版，将在亚马逊开发的云端运算服务环境 Elastic Compute Cloud (EC2) 上全面部署。MySQL 5.1 和高级消息队列协议 (AMQP) 也可以在新版系统中找到。

Canonical 还加强了与业界厂商的合作，在实验室中测试了来自 IBM、Sun、戴尔、惠普的 45 台流行中端服务器系统。

上网本的风行和 Windows Xp 的垄断让 Ubuntu 特地增加了一个专用版本，拥有更快的启动速度和更友好的用户界面，而且可以直接从官网上下载到 U 盘里，然后再安装到上网本里，相当于一步打造启动式 U 盘。

Ubuntu 9.04 Netbook Remix 已经在宏碁 Aspire One、华硕 Eee PC 1000、戴尔 Mini 9 等上网本上进行了全面测试，推荐尝试。

ChinaUnix 下载地址：<http://download.chinaunix.net/download/0013000/12433.shtml>

Moblin 项目易主 Linux 基金会全面接管

北京时间 3 月 31 日，非营利性机构 Linux 基金会宣布，他们将担负起 Moblin 社区的管理工作，而这之前，Moblin 社区一直是由 Intel 在管理和掌握的，这个社区也是 Intel 一手带大的。在这次接管中，Linux 基金会将会获得如下的权限，比如可以接触到 Moblin 的源码。

Linux 基金会主席 zemlin 在采访中提到，Moblin 是来自 Intel 公司的，Intel 是一个不缺乏资源和优秀天才的地方，而且他们也很看重 Moblin，他们把它看作一个战略平台，结合 Atom 芯片，用来和 Arm 处理器平台。将 Moblin 置身于 Linux 基金会的管理之下，才能得到更多的支持和更多开发者的青睐。Moblin 现在被使用在汽车电子、移动上网设备等轻型终端上，也许将来的某天，它也会出现在手机，甚至冰箱和电梯上。

我们并不能很好的预测出下一个极具创新性的事物的发生，但是可以肯定的是，将你的极具想法的念头和大家一起分享，他才能获得极大的成功。

Moblin 项目在 Intel 公司中占据很重要的地位，因为 Intel 一直都被微软等操作系统公司所恶心，不得不跟随他们的脚步而发展。在移动电子设备领域，Intel 再也不想做追随者，他们创立了 Moblin 项目，想借机拜托操作系统系统供应商的钳制。他们为此投入了大量的人力物力，比如从红帽 Linux 公司挖来了 Linux 业界著名的 Alan Cox。

同样，竞争无处不在，来自 Google 的 Android 和 Ubuntu MID 版本也在不断的蚕食 Moblin 的社区，Intel 为此也痛苦不已，所以才有了这次和 Linux 基金会的合作。在今年 6 月即将举办的台北 Computex 展会上，Intel 将带来 Moblin 最新的成果，联合 Genivi 联盟中的宝马、通用等汽车厂商，展出最新的车载移动控制设备，其中基础系统就是 Moblin。

虽然这次 Moblin 项目的管理权限已经到了 Linux 基金会手中，但是项目实际控制权还是掌握在 Intel 自己手中，因为 Moblin 社区中的主力开发人员都是 Intel 全职雇佣的，也许随着越来越多的开源开发人员加入其中，这种状况才会慢慢得到好转。

自由软件基金会主席对 Linux 没有采用 GPL v3 感到失望

GPL v3 许可证于 2007 年 6 月 29 日正式发布。两年之后，自由软件基金会主席 Richard Stallman (RMS) 在一次采访中谈论了自由软件和 GPL v3 的采用状况。

在谈到 Linux 时，RMS 说，我对 Linux 没有迁移到 GPLv3 而感到失望，GPLv3 为用户的自由加入了额外的保护，但它却没有应用到 GNU/Linux 系统的重要核心部分上。我有点害怕，因为我发现微软通过金钱收买某些项目去继续使用 GPLv2。尽管如此，GPLv3 仍然被许多自由软件项目所采用，甚至一些开发者用此许可证去开放发布以前的私有软件源码。总体上说，我认为 GPL v3 完成了使命。

在谈到 GNU 和 Linux 关系时，RMS 说 GNU 是类 Unix 的操作系统，它始于 1984 年。但 GNU 并不完整，它的内核从未能良好工作。所以我们多数使用的 GNU 是运行在 Linux 核心上。GNU/Linux 系统的出现是在 1992 年，自那以后，有许多程序加入进来，但 GNU 贡献最大。

MySQL 创始人宣布要创建 MySQL 分支

Sun 宣布发布了针对实时关键任务数据库应用的 MySQL Cluster 7.0。新特性包括增加性能和可扩展性，简化了集群 (cluster) 的备份和维护。

与此同时，去年离开 Sun 的 MySQL 创始人和开发者 Michael Widenius 在博客上表示，他有意

创建一个 MySQL 分支。他说，甲骨文公司收购 Sun 的理由是：Sun 制造硬件，虚拟机 VirtualBox，存储系统 Sun tape Storage，OpenSolaris，Java 和 MySQL。对于 MySQL，Oracle 可能会——“杀死它”，“卖掉它”，“通过开源社区强化它”。他个人最希望 Oracle 能选择第三种，但这意味着 Oracle 必须去学习开源开发模式及与社区通力合作，他对此不太乐观：Oracle 在开源界的名声并不好。所以为了保证 MySQL 能继续存在下去，他计划从正式版中创建一个分支——用他的公司 Monty Program Ab 去创建一个 MySQL “Fedora” 项目。

这引起关于谁拥有商业开源应用程序的讨论：是企业版权持有者，还是社区？有人认为从社区的角度去看的话，商标和版权都是毫无意义的。

Intel 投身社区 资助 GCC 编译器开发

三位 Intel 开发者开始参与 GCC 编译器的开发工作。到目前为止，加入 GCC 开发队伍的芯片制造商有 AMD 等，Intel 之前主要专注于自家的 Intel Compiler Collection 编译器。

大企业参加开源项目开发并不稀奇，IBM 有多位雇员全职参与 Linux kernel 和硬件驱动的编程工作。Intel 也已经向 X.org 项目捐赠代码，并维护着基于 Atom 平台的 Moblin Linux 社区（现在已经转交给 Linux 基金会管理）。

三位开发者之一的 Melanie Blower 已经发送邮件给自由软件基金会(FSF)要求相关的版权转让格式。除了 GCC 之外，Melanie 还计划参与 binutils，gdb(GNU debugger)和 glibc(GNU C library)的开发，她的编程经历可上溯到 ADA 语言。

GNOME 3.0 计划

GNOME 开发者 Vincent Untz 在其邮件列表中公布了 GNOME 3.0（亦即 GNOME 2.30）的计划。该计划着重谈到了以下三点：

- * Revamp our User Experience
- * Streamlining of the Platform
- * Promotion of GNOME

对于普通用户来说，可能会对第一点比较感兴趣。GNOME 3.0 作为一个具有里程碑意义的版本，大的变化是显而易见会有的。其中，GNOME 3.0 将引入两个新的核心组件：GNOME Shell 和 GNOME Zeitgeist。关于 GNOME Shell，我们此前曾向大家介绍过，这里不再赘述。单说 GNOME Zeitgeist，其主页说这是一个用于轻松浏览和查找文件的工具。有一段视频，可以让你直观的感受这个程序。

在 GNOME 3.0 之前，还有一个版本不容我们回避，这就是 GNOME 2.28。GNOME 2.28 预计今年 9 月发布，其 Web 渲染引擎将从 Mozilla/XulRunner 完全迁移到 WebKit。

根据 GNOME 六个月一个新版本的开发周期推算，GNOME 3.0 在明年（2010）三月有望与大家见面。

创始人：WordPress 已形成独特的经济生态圈

WordCamp 是全球 WordPress 使用者和开发者们的一个重要聚会，目前已在美国、以色列、

中国等国家的多个城市成功举办，WordCamp China 2009 于 4 月 4 日在上海举行。

WordPress 已形成经济生态圈

在最近更新的版本中，WordPress 现在拥有了一键更新功能，用户无需掌握高深的 PHP 代码技巧便可简单顺利地将 WordPress 版本升级到最新。最新的 2.7.1 版本每天的下载量超过 32000 次，这个数字还不包括各种分分流站点的下载量。

经过了几年的发展，如今使用 WordPress 平台的博客越来越多，已经建立了一个独特的经济生态圈。据马特在旧金山 WordCamp 的现场调查显示，超过三分之二的与会者在为其他公司提供与 WordPress 平台相关的商业咨询服务。

自由软件社区的巨大贡献

最初的版本，操作后台十分简陋而且没有任何的色彩。经过开源社区四年的努力，WordPress 已经改头换面变成最时尚、最漂亮的独立博客系统之一。借助自由软件社区的力量，大约有 150 名开发者为 WordPress 2.7 版代码作出过贡献。这样数量的开发者，已经相当于一家中等规模的软件公司。

关于 WordPress 2.8

在 2.7 版中，WordPress 增加了内核升级和插件升级两个深受用户欢迎的功能。在下一版中，升级功能将会覆盖到博客主题(Theme)，使 WordPress 系统更加便于使用。为了能让非英语用户可以更好地使用，将会增加插件和主题的国际化功能。在不久的将来，所有的流行插件都将同时拥有包括中文在内的多语种版本。

顶级脚本语言：PHP、Ruby、Python 之争

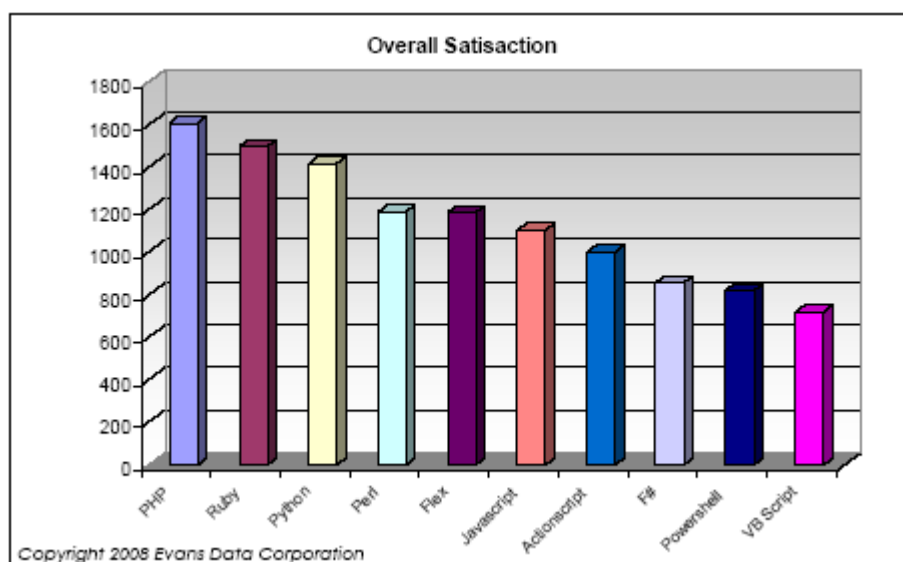
Evans Data 发布了[脚本语言调查](#)结果。在 500 多名开发者和 IT 专家心目中，PHP、Ruby、Python 是他们最爱的语言。随着时间推移，企业中脚本语言的应用也发生着变化，这份调查列举出了每种语言的优缺点。Richard Dice ([Perl](#) 基金会主席) 在近期的[采访中](#)表示，脚本语言在企业里面正在呈现上升趋势，不仅仅用于简单任务或是原型构建。它们的应用范围更加广阔。

在调查中，用户要根据以下种种特性（源自调查问卷）给语言评分：

- 易于使用
- 异常处理
- 扩展
- 可维护/可读
- 跨平台移植
- 社区
- 可用的工具
- 工具质量
- 性能
- 内存管理

- 客户端脚本
- 安全

所调查的语言包括：Actionscript、Flex、Javascript、Microsoft F#、Microsoft Powershell、Perl、PHP、Python、Ruby、VB Script。最终根据用户的意见生成调查报告。综合评分（包括权重）将 PHP、Ruby、Python 推到了首位。



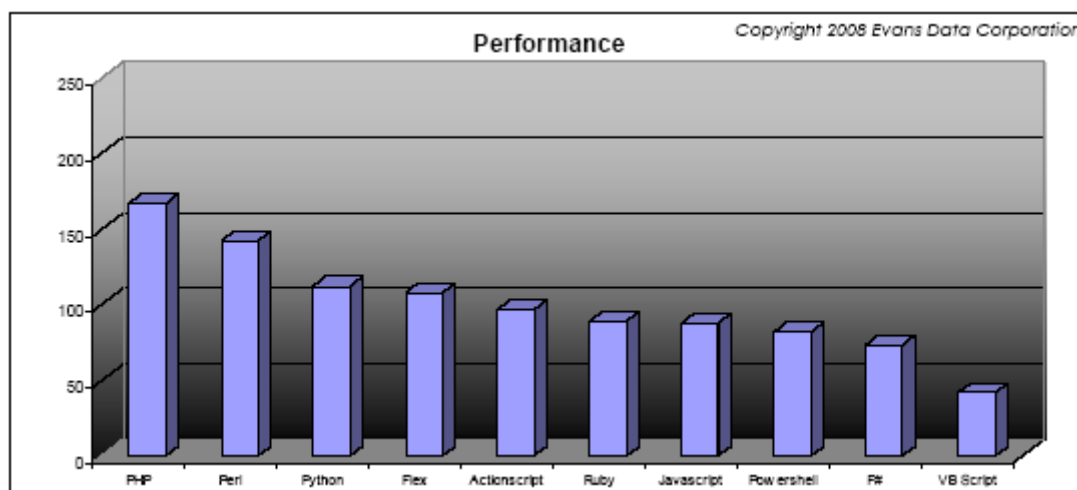
在这次调查中，最优秀的语言是那些开源语言，它们以符合自然发展规律的方式进化。而那种私有的 Microsoft 语言，虽然背后有着软件巨头的强力支持，也有众多资源，但并未像开源语言一样受到用户好评。

在用户眼中，Ruby 和 PHP 的易用性最好，Python 的扩展性则评分最高。在这三种语言中，Ruby 的社区力量尤为显著：

Ruby 社区相当小，但每个人都积极参与，勇于表达自己观点。

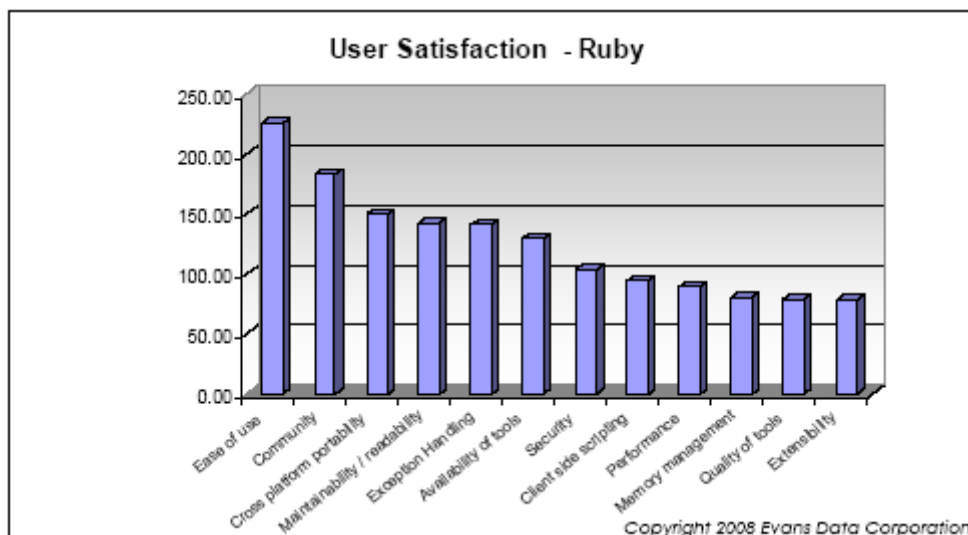
报告中也提到了一个常见的争论：Ruby 的性能。

即便是最热衷于 Ruby 的用户也会对 Ruby 应用的性能有所怀疑，而且有些广为人知的用 Ruby 编写的 Web 2.0 应用，性能和扩展性方面已经暴露出了很严重的问题。



报告中也突出了另一个问题：开发者对安全的关注还是一如既往的低。

最后来看，Ruby 的评分从高到低分别是：易于使用、社区、跨平台移植、可维护/可读、异常处理。



[Antonio Cangiano](#) 和 [Peter Cooper](#) 对 09 年中 Ruby 面临的挑战持异议，他们认为，Ruby 面临的真正挑战来自于技术或是传道。EDC 调查报告显示，Ruby 平台需要在客户端脚本（这方面 JavaScript 的得分最高）和工具（质量和可用性）。

传谷歌拟削减对火狐支持 挖角 Mozilla

据国外媒体报道，消息人士表示，为了更好的拓展浏览器业务，谷歌计划大幅削减对火狐 (Firefox) 开发商、非赢利性机构 Mozilla 的支持，此举甚至引发多名 Mozilla 工程师有意跳槽谷歌。谷歌不仅想要称霸搜索引擎市场，还希望拓展浏览器业务，这也是它去年秋天发布 Chrome 浏览器的原因所在。但那时也只是开发了一款与火狐对抗的产品而已，但如今，谷歌却准备压缩火狐的钱袋子。

据国外媒体报道，消息人士表示，为了更好的拓展浏览器业务，谷歌计划大幅削减对火狐 (Firefox) 开发商、非赢利性机构 Mozilla 的支持，此举甚至引发多名 Mozilla 工程师有意跳槽谷歌。

谷歌不仅想要称霸搜索引擎市场，还希望拓展浏览器业务，这也是它去年秋天发布 Chrome 浏览器的原因所在。但那时也只是开发了一款与火狐对抗的产品而已，但如今，谷歌却准备压缩火狐的钱袋子。

尽管谷歌发布了自己的浏览器，但它仍然为 Mozilla 提供资金支持。Mozilla 与谷歌达成了搜索推荐协议，当用户在火狐浏览器的工具条内进行搜索时，谷歌就会向 Mozilla 付费。Mozilla 基金会有 90% 的收入都来自于这一业务。而就在谷歌发布 Chrome 浏览器之前，Mozilla 还刚刚与谷歌续签了协议，将其延长至 2011 年。

但关注谷歌的消息人士却透露说，谷歌正在探索多种方式，希望大幅削减对 Mozilla 的支持。这也使得 Mozilla 的工程师惶恐不安。据悉，目前已经有多名 Mozilla 工程师来到谷歌总部面试，希望加盟谷歌。

业内人士认为，谷歌想要支持自己的 Chrome 浏览器无可厚非，但对 Mozilla 这样一家非赢利

性机构下手却似乎有违其“不作恶”的信条。由此看来，在如今的情况下，“不亏钱”要比“不作恶”更加重要。

开源社区争论话题之二：走向歧路的标准组织

怎样才能制定出一个真正的技术标准？是否需要国际标准化组织反复检查、争论和投票？如果它只是厂商锁定的一个橡皮戳怎么办？Linux 之父李纳斯·托沃兹(Linus Torvalds)表示，真正的标准应该从基础开始制定。本篇文章是《开源社区争论话题》系列文章的第 2 部分。在第一部分中，我们介绍了围绕 Linux 内核发展的争论；第二部分我们就看一下开源社区对标准化过程的看法。

去年当微软成功让其 Office Open XML(以下简称 OOXML)摇身变为一个国际标准的时候，对于开源社区内外的很多人来说，这是一个非常重大的事件。

OOXML 成为国际标准的整个过程非常令人生疑，它遭到了全球很多国家和企业的反对，但是国际标准化组织最终的决策却让很多人感到震惊、怀疑、甚至是愤怒。

对此事的嘲讽之声一直持续到今天，我们还可以听到人们在争论，标准化过程是否从根本上已被完全破坏。所有争论的一个中心问题是：标准是否已经变成一个实现厂商锁定的武器，或者开放性是否依然有机会获得保障？

标准组织已经成为政客和商人的舞台



Linux 之父托沃兹表示，“我一直在远离标准化组织，因为那里是卑鄙小人和政客们聚集的地方，我真的不想与它们有任何关系。”对于 OOXML 成为国际标准一事，托沃兹表示，“与此有关的所有人一定是疯了，一切都那么令人厌恶。”托沃兹断言，这样的标准肯定不会成为最重要的标准之一。

托沃兹解释称，“我个人强烈认为，往往真正有问题的标准是那些所谓的事实标准，即先成为业界事实标准然后再被认可，而不是提前成为标准。这让我想起了最初的 POSIX.1 标准，事实证明它是一个非常好、非常成功的标准，但这是因为它主要是对先有实践改造后才落实成标准，而不是试图创建一个新标准。”

对于开放性和标准之间的交叉点，托沃兹表示，“开放性最终会孕育标准。”不过，托沃兹没有花费太多时间来担忧这个问题，他表示，“我的工作是在制定最好的技术，或者更准确点说，是开发那些能够给人们带来更好体验的技术。我不会因为标准组织和厂商政治而自寻烦恼。我的精力没有那么多。”

真正标准 VS “强盗” 标准

《精通 vi 和 Vim》一书的联名作者艾尔波特·汉纳(Elbert Hannah)表示, 一个真正的标准是“开放、自由、易于使用和修改的。”如果不严格的来说, 目前的例子包括 MP3、TCP/IP、vi/vim 和 POSIX。

另一方面, 汉纳也列举了几个名为标准实则“强盗”的例子, 诸如 Word、Windows、DOS、英特尔芯片和 HD DVD 类标准。他解释说, “以上标准可能每个人都会使用, 但是它们并非因为自己的优点而被广泛采用, 而是凭借自己强大的市场份额, 它们无所谓优秀或糟糕, 它们没有经过投票表决。”汉纳表示, 并不是要求任何标准都不能有所有者, 但是有一点应该是肯定的, 即使有人拥有一个标准, 他们也不应该对该标准具有绝对的权力和控制, 例如它们不能命令全世界来如何使用这个标准。

目前标准似乎主要用来封锁客户到某个厂商中, 微软的 OOXML 标准争议就是一个很好的例子, 汉纳表示。微软声称有意使其成为一个开放、统一的文档标准, 但是其真实目的是, 这是一个严重倾向于它自己软件和平台的标准。汉纳表示, 这不是标准的真正用途。他认为, 标准最原始的模式更可取, 即发现需求, 对其解决, 然后让它被广泛应用。这个过程很重要的一点是“真实”, 它必须不受任何来自第三方利益的影响。

但是不幸的是, 今天的巨无霸厂商在通过所谓的“标准”来损害开放和友好的技术。Slashdot 博客马丁·埃斯皮诺萨(Martin Espinoza)表示, 尽管标准有很多优秀的地方, 但是它们的问题是具有强制性。

不过, 埃斯皮诺萨指出, Office 文档是一个事实标准, 之前已经有一个国际标准。ISO 的做法实际就是立法腐败。总体来说, 人们不会注意到非常大的谎言, 但是像这种相对较小的谎言却十分令人厌烦。来自蒙特利尔的开源顾问格哈德·麦克(Gerhard Mack)表示, 从任何角度讲, 微软或许赢得这一场战役, 但却失去了整个战局。

麦克解释说, 微软认为它们将拥有一个新标准, 就可以在争取用户支持的竞争中领先很长时间, 但事实并非如此。

标准化组织需迷途知返

相反, 麦克认为, OOXML 标准在标准化过程中被进行修改, 因此目前市面上还没有支持它的产品, 包括微软自己的 Office 2008。麦克补充说, 这当然是微软的障眼法, 目的是为了迎合标准委员会的喜好以及消除来自外界的反对之声。至少从现在看来, 标准制定过程阻碍了技术开放。麦克表示, “不幸的是, 标准已经成为企业扩大自身优势的战术武器。在微软推动 OOXML 标准化的过程中, 以及 Rambus 的标准化中, 我们看到了这种意图。”

麦克表示, 展望未来, 标准化组织需要坐下来承认存在的问题, 然后决定是希望帮助还是阻碍行业发展。他表示, “我希望 ISO 组织能够改正错误, 它们已经遭到广泛的指责, 但却拒绝承认任何错误。”

麦克建议, 该组织不仅仅要重新考虑自己的快速审批过程, 而且需要制定一个标准来确定成员国如何选择代表。否则常此以往, 它们可能会发现自己不再被认可, 被那些需要标准来真正孕育互通性的人所拒绝。

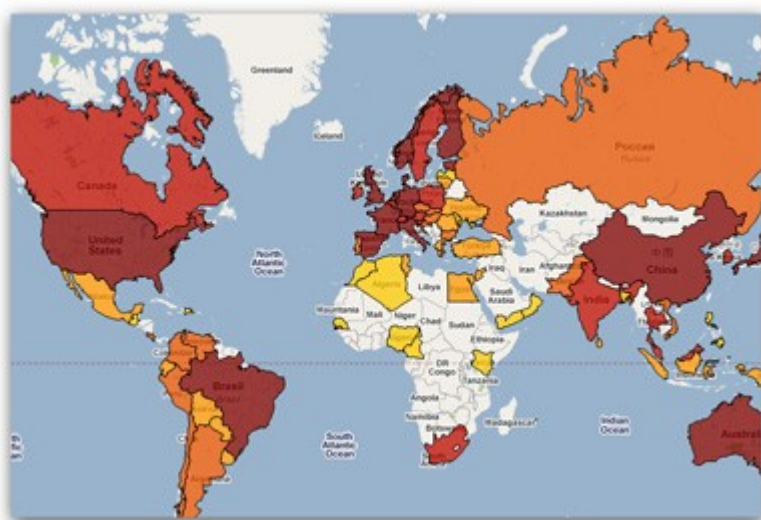
微软资助密歇根州立大学开发 NFS v4 windows 版本

微软最近和美国密歇根州立大学达成协议，前者将为该大学的信息技术综合研究中心（CITI）提供资金，来开发开源版本的 NFS(网络文件系统)v4 版协议，后者将被用在 Windows 操作系统上。微软已经在 Windows 桌面版和服务器版中提供 NFS 支持，但是目前仅提供第三版支持。目前该大学的研究中心也正在为 Linux 开发 NFS v4。

第四版的 NFS 将加入一些新特性，包括支持文件锁定和大量协议，更高的安全性，支持复合化操作，客户端缓存等。微软为 CITI 提供的资金是用来开发 NFS v4 的 Windows 端口，微软表示，“CITI 可以拥有自己的时间表，开发和发布计划，当它完成之后我们的客户就可以获得它”。

这就是说，NFS v4 的 Windows 端口开发计划不是由微软决定的，而且目前还不清楚微软将把 v4 加入到哪个版本的 Windows 中。

Red Hat 公布“开源世界地图”



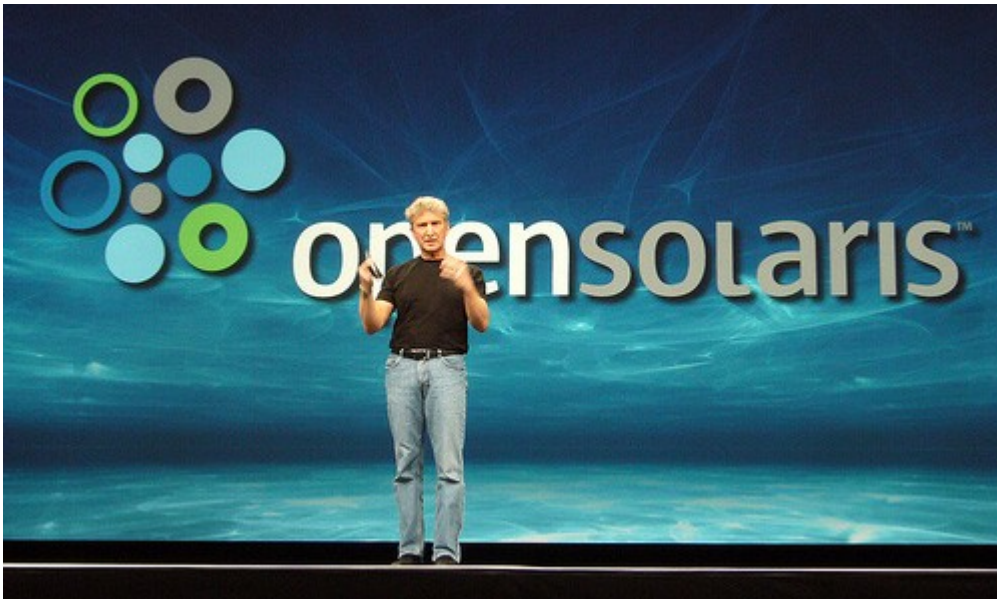
RedHat 和乔治亚理工学院合作绘制了 75 个国家的开源活跃性地图，地图正式名称叫开源指数 (OSI)，依据一系列因子包括政策、政府实践、行业和社区统计得出最终排名。前 16 名如下法国：1.35(得分)；西班牙：1.07；德国：1.05；澳大利亚：1.04；芬兰：1.03；英国(英格兰，苏格兰，威尔士和北爱尔兰)：1；挪威：0.95；爱沙尼亚：0.89；USA：0.89；丹麦：0.79；意大利：0.71；巴西：0.68；荷兰：0.67；日本：0.62；中国大陆：0.6；新加坡：0.58.....

结果有点令人意外，中国竟然属于开源高活跃性国家行列之中，与通常的观点相去甚远，中国排在比利时、加拿大、澳大利亚、瑞士等发达国家之前。

行业观察

甲骨文收购 Sun 或整合 OpenSolaris 和 Linux

4月21日消息, 业内分析师称, 一旦完成收购 Sun 的交易, 甲骨文可能整合 OpenSolaris 和 Linux, 但不会放弃被广泛使用的 Solaris 操作系统。



据国外媒体报道称, 甲骨文董事长拉里·埃里森(Larry Ellison)表示, 该公司收购 Sun 的原因之一就是其基于 Unix 的 Solaris 操作系统。长期以来, Solaris 一直是甲骨文数据库运行的主要平台, 用户群相当庞大。

甲骨文还是 Linux 的热心粉丝, 因此如何处理两种操作系统之间的关系颇受外界关注。事实上, 就在两周前, 甲骨文首席架构师爱德华·斯格列文表示, 甲骨文将把 Linux 作为其数据中心的首选操作系统, 因此客户也不要再选择其他操作系统了。

市场研究公司 Ovum 分析师大卫·米歇尔表示, 甲骨文不可能让 Linux 取代 Solaris, 因为 Solaris 能够带来稳定的维护业务营收以及服务营收, “这些业务的利润率在 90%以上, 甲骨文不会损害该业务。”米歇尔预测, 甲骨文将继续投资和升级 Solaris。但是, 甲骨文可能会在 Linux 中整合 OpenSolaris 的部分功能。

市场研究公司 RedMonk 的分析师斯蒂芬·奥格雷迪表示, 整合 OpenSolaris 和 Linux 的一大障碍是两者的许可协议。OpenSolaris 采用了 Sun 的“通用开发和发行许可”(CDDL), Linux 则采用 GNU 的“通用公共许可”(GPL)。甲骨文可能改变 OpenSolaris 的许可协议, 使 OpenSolaris 和 Linux 可以互通有无。OpenSolaris 中的某些功能对于 Linux 非常有用。

米歇尔表示, 人们一直期待 Linux 能拥有 Sun 的 DTrac 技术。DTrac 使系统管理员能够在系统运行期间解决内核和应用软件出现的故障。Containers 技术也可以在 Linux 中找到用武之地, Containers 是一种类虚拟技术。

开源风投：回报时刻即将到来

北京时间 4 月 14 日消息，开源内容管理软件开发商 Alfresco 业务开发副总裁马特·阿塞(Matt Asay)周一称，据市场研究公司 The 451 Group 最新公布的报告显示，1997 年至今，风险投资者已经向开源公司投入了 32 亿美元资金，这些投资目前已经到了开始获得回报的时候。

阿塞称：“当然，这 32 亿美元的投资已经通过某些方式而获得过几次回报。举例来说，Linux 基金会预测，免费研发的 Linux 工具包价值为 108 亿美元。此外还有一种观点十分引人注目，即开源软件开发商已经为客户节省了 600 亿美元的许可费。实际上，如果不光是计算开发商首创的开源软件，那么这些软件的价值还将远高于此。”

但阿塞指出：“虽然这很不错，但风险投资者进行投资的目的是自己获得回报，而不是 IT 企业或发展中经济体获得回报。除了 Red Hat、Suse、XenSource、Zimbra 和 JBoss 等少数开发商以外，开源软件的生态系统一直没能‘养肥’风险投资者，这种情况必须得到改变。”

阿塞表示：“我相信，我们已经站在了峰顶，改变即将发生，原因在于：

在我们公司的上次管理会议上，我将自公司 2005 年 11 月首次出售产品以来的销售表现与道琼斯工业平均指数(以下简称“道指”)进行了对比。如上表所示，头两年里 Alfresco 销售追随道指走高，而在过去一年中，道指急转直下，而 Alfresco 销售则一路飙升。

对开源软件开发商 Alfresco 来说，经济衰退一直都是件很好的事情，但不只是我们如此。我已经与 SugarCRM、JasperSoft、Volantis 和 Openbravo 等由我提供咨询建议的开源软件开发商进行了对话，同时也联系了 Sun's MySQL、Pentaho 和 OpenX 等我不提供咨询建议的开发商，结果发现几乎每家开源软件公司都存在着一个共同点，那就是——经济下滑，销售上涨。

我预计，经济将在 2009 年末或 2010 年初开始略有复苏，从而提供并购交易的估值。一旦发生这种情况，那么我相信风险投资者将开始从其开源投资中收获巨额回报。”

IDC：2009 年 Linux 软件销售收入将增 21%

4 月 11 日消息，据市场研究公司 IDC 最新发表的题为《Linux 在新经济中的机会》的报告称，Linux 的销售收入将显著增长。不过，大部分增长并不是以微软的损失为代价的。

IDC 的报告称，用户 2009 年的 Linux 开支预计将比 2008 年增长 21%，超过整个软件市场的增长速度。整个软件市场 2009 年的增长率是 2%。2013 年全球 Linux 市场的销售收入将从 2008 年的 123 亿美元增长到 355 亿美元，复合年增长率为 23.6%。

尽管如此，重要的是需要指出 Linux 与 Windows 市场规模的比较。2008 年微软 Windows 的市场规模是 1490 亿美元。微软指出，即使微软从现在至 2013 年的增长率不到 10%，2013 年 Windows 的市场规模也将增加 560 亿美元。

IDC 认为，虚拟化是推动 Linux 增长的一个重大因素。云计算也将推动 Linux 和开源软件的应用。真正的收入将来自于 Linux 的应用。Linux 在 ERP、数据库等传统的工作量中的应用将增长 13%至 18.6%。Linux 在传统工作量方面的增长主要是以 Unix 应用的减少为代价的。

IDC 的调查发现，53%的企业正计划增加 Linux 在服务器中的应用。48%的企业预计增加 Linux 在客户机（台式电脑和笔记本电脑）方面的应用。

让开源软件真正用起来在于执行力

作者：胡才勇

曾经有一段时间，许多人都认为开源的希望在中国。特别是在 2001 年的时候，中国开展的第一次软件正版化政府采购在北京进行，全面采购了基于开源软件发展起来的国产基础软件，包括国产 Linux 和国产办公软件，微软产品全部出局。这一事件在全球引起很大震动，当时许多人更是认为，在桌面领域打破微软垄断的希望就在中国。此后，我国政府继续通过政策、财政和科技攻关项目等加强对开源软件的支持力度，经过几年的发展，我国产 Linux 操作系统和 Office 办公软件在技术上已经具备了替代微软操作系统和办公软件的能力，但实际推广情况并不乐观，特别是实际的使用情况。这问题到底出在哪？

在中国，linux 操作系统等国产基础软件的推广一直受困于用户多年的“使用习惯”和事实上的大面积盗版，留给国产基础软件的市场极为狭小。即使是在政府领域，作为支持开源软件的最重要力量和战场，在公用财政支持的采购领域，也未得到真正的大面积应用。相反，国际上一些国家却已实实在在地把 Linux 用了起来。

在挪威，大面积采用开源软件首先开始于教育部门，越来越多的学校全盘采用开源软件，此后不断向其他领域扩展。挪威政府此前还公布一项政策，要求所有公共部门在 2006 年前必须制定采用开源软件的计划。此外，还要求所有公共部门的 IT 系统在 2009 年前全部采用开放标准。挪威政府对开放标准的支持力度直接促进了国内采用开源软件的热情，目前，全国上下向 Linux 大规模迁移的计划正在进行中，在许多城市正逐步地将市政管理的所有桌面迁移到开源操作系统上。

在巴西，推广开源软件经历的是一种“自下而上”的路线，在州和市政府不断加大推广和迁移步伐的基础上，巴西联邦政府开始全国性的推广策略。可以说，是州和市政府的推广经验首先为全国的大规模推广铺平了道路。据媒体报道，目前，巴西 22 个联邦部门中已有 7 个在使用开源软件，包括许多开源桌面配置，其中有至少 4000 台主机使用 OpenOffice.org。最近，巴西政府正在进一步研究财政方案，考虑制造二百万台预装 Linux 的低成本笔记本，并将其中的 100 万台分发给地方学校使用。

在法国，国家宪兵部门和准军事警察部队逐步开始弃用微软的操作系统和办公软件，转为使用开源软件。在美国，最新出现的对开源软件的扶持政策是美国纽约州立法机构修改美国税法，对开源软件开发者进行个人所得税补贴（减免 20%），最高补贴限额为 200 美元/每年。在美国的一揽子经济刺激计划中，提到了要促进开源软件在医药部门的应用，这对开源软件在美国的发展也将产生直接的促进作用。

在俄罗斯，此前盗版率高达 99%，同样深切面临打击盗版和发展国产软件的压力，最终，俄罗斯政府以加强开源软件发展作为打击盗版的道路。从 2007 年起，俄罗斯在所有学校部署采用开源软件，并在各地举办各种学习班开展 Linux 培训。（以上例子摘自媒体报道。）

.....

其实，认可开源软件能用的人非常多，但认为开源软件“好用”的人就相对要少许多。在我国，所谓的好用，其标准就是微软。说微软就是好用的标准，显然是不科学的，但习惯力量就是这样认为。

挪威从教育开始，巴西是从试点开始走向政府的统一行动，这些举措我们其实也开展多年，但并未实现最终的全国性的推广成效。我们在教育领域已有试点，在电子政务系统领域试点更是非常多。最典型的如全国 40 所重点高校 Linux 实验室、北京市平谷区圈国产化的解决方案、湖北荆门掇

刀区的电子政务系统、中联部的全国产解决方案等都是采用 Linux 的成功试点。但在大面积推广时，我们确实举步维艰。理由听起来都很充分。

是国产软件没有准备好了吗？可是，如果不能大面积的使用，国产软件永远不可能准备好。是我们没有相关的推广政策吗？国内支持国产软件的各种政策已很多，关键是在执行这一步，完全在于执行者的力度。我们在推广国产软件的过程中，明显地感到了“人治”的特色。同样的政策，遇到不同的人执行的效果就不一样。每一个成功的试点后面，都有一个强力推行国产软件的“第一责任人”。这种状况，在试点阶段是可以的，但要大面积推广则不行。

一些专家学者已经呼吁，促进开源软件的推广和实际应用关键在于政府的推广决心。在我国，虽然对于开源软件的支持一直在进行，但确实缺乏全盘推进的决断，无论是在教育领域还是在电子政务领域，都尚未制定和实施具体的大规模推广计划。在美国、在欧洲、在巴西，人们都认为为用户提供更多选择，以开源软件的推广来打破垄断是最佳方式。在我国，电子信息产业振兴计划和重大专项都在努力引导民族产业的发展。希望产业的大规模推广是我们下一步的重点。

去年的黑屏事件，再次凸显一个残酷的现实：操作系统和办公软件的绝对市场优势，使垄断者具备了操控用户的能力。因此，无论是从经济角度出发，还是从信息安全保障而言，都需要有关部门加大推广决心，加大对开源软件的支持力度，让更多的部门把开源软件和国产基础软件真正用起来，也将是促进我国软件产业全面发展的出路所在。

开源软件开发成本高达 3870 亿美元

北京时间 4 月 15 日，Black Duck 发布文章称，开源软件开发成本估算在 3870 亿美元。这个数字很吓人，已经到达了美国救世计划的资金 7870 亿美元的一半还要多。

Black Duck 给出的数字，基于其掌握的大量的开源软件信息，其官方网站刊登的文章分析指出，超过 20 万的开源软件代表着超过 49 亿行的代码，开源软件全部开发成本超过了 3870 亿美元，以及相当于共同投资了每年超过 200 万的程序员。

另外分析指出，10% 的 IT 应用开发花费是多余的，美国的公司应该认识到再利用开源软件在程序开发中的重要性，通过它可以来每年节约 220 亿美元。它具备成本、时间和机会的优势。

Black Duck CEO, Timothy Yeaton 预测：敏感的企业将利用开源软件创新，达到重复投资的节约，从而获取竞争优势。通过企业允许开发者为工作寻找并使用合适的开源软件，包括开源软件与其他代码的整合，将使开源软件成为引人注目的建议和选择。

Black Duck 的分析观点，与欧盟的一项调查报告《2006 study by the European Commission's Directorate General for Enterprise and Industry》的结论相同，2006 study 报告认为再生产已经存在高质量的开源代码，将消耗欧盟企业 120 亿欧元。

笔者认为 Black Duck 的数据 3870 美元有些夸大的味道，但是也说明了当前的经济危机期，企业应该充分利用开源资源。开源软件在现在以及以后会更有活力，更能体现价值！

微软 TomTom 和解案留下许多问号

微软与 TomTom 和解，外界反应不一，但似乎有个共识，就是有关 Linux 是否侵犯微软知识产权的问题，仍然悬而未决。Pillsbury Winthrop Shaw Pittman 法律事务所知识产权部门领导人 James Gatto 律师说，此案快速和解，可能是 TomTom 在评估微软案的利害关系之后，所做的务

实商业决定。

繁复的诉讼，特别是本案牵涉的专利诉讼与反诉，动辄要耗费 1000 万到 1500 万美元的诉讼费。Gatto 说：就微软的专利是否涵盖 Linux 而论，我不认为问题得到解决。微软长久来宣称，Linux 的许多实作都侵犯该公司的知识产权，并已和多家散布或采用 Linux 的公司敲定一连串的专利协议。

但 TomTom 案却是微软首度告上法庭。开放源代码先驱 Bruce Perens 批评这桩和解案，指出这可能让其他使用嵌入式 Linux 的公司心生畏惧，进而产生寒蝉效应。现任开源软件开发公司 Kiloboot CEO 的 Perens 说：我的感想是，公平正义并未彰显。微软(在 TomTom 案)的专利称不上创新，法庭也许会认定它们无效，但 TomTom 却被迫为这些专利付钱。他们不花钱证明专利无效，因为或许负担不起庞大的官司诉讼费，因此 TomTom 决定付钱取得专利授权了事。

Gatto 说，微软对 TomTom 采取的行动，未必意味微软准备对 Linux 开战。他说：微软过去几年来变得更积极主张自己的专利权，这不只是冲着 Linux 而来，而是全面性的。高分贝表示不和微软缔约的 Linux 公司 Red Hat 发表声明说：缺乏司法判决，这起和解案并不显示微软的主张有效。专利诉讼是棘手的过程，除了是非曲直之外，还有许多因素，可能促使 TomTom 这样的被告在目前这种经济环境下，选择和解。

Gatto 指出，微软处理 Linux 问题一直小心翼翼，避免与整个开源社区为敌。至于开源公司能从此案获得什么教训，Gatto 表示，所有的企业都必须了解与自家产品有关的知识产权问题。他说：专利遍布各行各业，软件业不例外，开源也不例外。开源并未受到比较大或比较小的关注。当前每一家公司都必须在知识产权方面提高警觉，不论是攻还是防。

和解案引发的第二个问题是，微软与 TomTom 是否真的能遵行和解协议中的通用公共授权 (GPL) 条件。双方未公布详细的和解条件，只表示约定不会控告 TomTom 的客户。Red Hat 说：据我们所知，和解授权未对外公开，所以我们无法评论他们能不能遵照开源要求与原则行事。

《知识产权与开放源代码》一书作者、Haynes and Boone 事务所律师 Van Lindberg 说：微软或许不希望 FAT 专利在法庭上接受检验，许多人担心面对高昂的专利官司。TomTom 则或许担心面临长期、所费不赀的专利诉讼战，而且引起争端的技术并不是他们产品的核心技术。Lindberg 指出，微软设法让协议符合开源合约的条件。他说：显然微软已做出结论，就是 Linux 以及 GPL 是市场的参与者，自有一套运作的规矩，必须照章行事。

Lindberg 说，他觉得此案最有意思的一点，是 TomTom 决定在两年之内，从自家产品中移除 FAT (文档配置表)。他说：乍看下，这似乎是微软的胜利，但长期而言反倒是会降低微软的专利使用率，如同 GIF 专利之于 Unisys。同时，软件自由法律中心(The Software Freedom Law Center)宣称 TomTom 案以和解落幕，是一大胜利。

该组织发表声明说：微软与 TomTom 和解，显示本社区反抗微软专利侵略的力量。TomTom 避免冗长、代价高昂的诉讼，且会持续是 Linux 导航设备的世界领导者，会继续凭自己与社区的力量创新，以免使用无效的专利妨碍本社区制作软件的能力。

又一开放精神与既得利益的拉扯战争

即便是在我们认为非常开放的 Google，总还是因为某些关系，而不得不有些限制。当然，Google 所做的限制，不能说对或是错，正如 moco 所说，就是一套游戏规划。这件事说来话长，我们可以从 Sam 的程序分享来开始谈。

Sam Lu 与杰森，他们花了一段时间，将 GWiFi 改版，并称之为 aNetShare 软件。它的功能就

是可以把 Gphone 手机当成无线基地台，然后分享无线网络，给你或你的朋友桌上型计算机、笔记型计算机联机上网。

但在四月一日的时候，收到 Google 来信，说「Root Hack」相关程序都必须下架，而无线分享程序也影响到 Google 的政策，当然也要下架。

而这件事情自然引起 Open Source 界的朋友很大反应，他们开始了行动，希望能够让 AndroidMarket 维持开放精神，因此出现了一套「Keep Android Open Petition!」，就是让 AndroidMarket 上面不要禁止「未授权软件」的呼吁，并且可以方便的让大家 download Android wifi 分享程序，结果反应热烈。

而 Google 也迅速对于这件事情做出处理，Sam 也贴出 Google 的说法，简言之，就是 Google 决定让 T-mobile Us 之外的人，仍然可以存取。

但到目前为止，aNetShare 这个程序仍未在 Market 出现？

Oracle 需要的是 Zend 而不是 Red Hat

首先，甲骨文已经表明他可能会采用 Red Hat 的产品并通过开源许可条款为其更改名称。



甲骨文企业 Linux 基本上就是 Red Hat 企业 Linux，只是贴上了 Red Hat 的商标而已。甲骨文不需要因为软件而购买 Red Hat，因为，Red Hat 的软件是开源的。

甲骨文可能只是想从 Red Hat 获取客户和支持而已，但是考虑到 Red Hat 是许多供应商的战略平台，又似乎不是那么回事。

另一方面，还有另一个战略性的开源供应商商业 PHP 供应商 Zend，可能很适合甲骨文。

甲骨文和 Zend 已经在 PHP 的分配上有合作，所以，他们对彼此并不陌生。甲骨文参与了广泛的 Java 生态系统，但是如果它收购这一主要的 PHP 供应商，那甲骨文将在 PHP 领域获得领导地位，也会为其自身开拓出一个真正的市场。PHP 是 LAMP(Linux/Apache/MySQL/PHP)网络堆栈中的一个关键组成部分。有了 Zend，甲骨文就可以很好地控制自己的网络堆栈业务了。

Zend 在私有的开源供应商中是一颗闪亮的明珠。其技术得到了广泛的应用，但是笔者似乎还没有从类似

这样的大宗交易中看出到底有多大的收益。甲骨文有资金和开源的专业知识，对于他们来说关键是要选择可以为自己的平台增添附加值并有助于扩展其平台的交易对象。

惠普称正在评估 Android 尚未敲定是否采用

4月3日消息，惠普正在研究是否采用谷歌的 Android 操作系统，但还没有决定是否在其产品中应用这款操作系统。

据国外媒体报道称，目前该款免费开源 Android 操作系统用于智能手机上，但其设计宗旨是支持所有上网设备，许多分析师视 Android 为上网本平台。Android “登陆” PC 都将对微软 Windows 操作系统的主导地位形成挑战。

尽管没有一家 PC 厂商公开承诺，包括微软在内的许多业界巨头都预计市场上将出现 Android 笔记本电脑。惠普一名女发言人表示，“该公司只是在研究 Android，以充分理解竞争对手所采用的各种操作系统或对我们的客户有价值的操作系统。我们希望评估 Android 在计算和通信方面的性能。”

华尔街日报援引消息人士的话报道，惠普正在测试一款可能推出的 Android 上网本。惠普目前发售的上网本运行的是 Windows XP 和开源 Linux 操作系统。

手机 Linux 安全之路 如何免于步履蹒跚

有人说，手机操作系统正在上演着 Windows Mobile、Symbian 和 Linux 之间的三国演义，而决定胜负的关键在于各自能够获得多少关键性角色的支持。Windows Mobile 当然有微软在支持，Symbian 背后是诺基亚，而 Linux 显然已经得到了 Google 的加持。不过，与前两者不同，Linux 拥有开放源码社区的背景，不仅可通过免费获得源代码而大幅降低手机制造商和软件开发商的成本，而且还可以得到丰富的社区资源，因而吸引了众多的参与者。

道路并不平坦

虽然 Linux 对于手机开发商充满吸引力，但这条路也并不平坦。例如实力雄厚的摩托罗拉就在 Linux 手机上遇到了不少麻烦。利用 Linux 来开发手机，遇到的最大麻烦其实也是由其最大的吸引力造成的。在开放源码社区里，大家都可以贡献代码，虽然资源因此而丰富，但却带来两大问题：

1. 怎样才能找到高质量且安全可靠的代码？
2. 怎样把各种功能代码模块集成在一起？

为了解决好这两个问题，Google 牵头成立了 OHA(Open Handset Alliance，开放手机联盟)，同时 LiMo 基金会也汇聚了众多的手机巨头。OHA 和 LiMo 虽然存在某些竞争，但其核心目标都是以 Linux 为基础建立手机软件标准化平台。有心人会发现，在这两个 Linux 手机行业组织中都有风河(Wind River)公司，而且是其中唯一的 Linux 平台软件提供商，而其他的成员大多是手机制造商、移动通信运营商、硬件或者软件部件厂商。而且，风河公司最近还被 LiMo 选为该组织的官方系统集成商，为这个目前集结了最多手机厂商的基金会提供通用基础架构、工具和集成服务。

前景十分诱人

有市场研究机构的分析师指出，随着功能创新的需求不断增加，软件在手机总成本中所占的比例也随之增加。在智能手机中，软件在 BoM(Bill of Materials，物料清单)中所占的比例已经超过 25%。特别是对于采用开源软件的手机厂商来说，软件集成显得尤为重要。就算是从开放源码社区免费得到了软件，花费在验证、集成和测试等工作上的时间和成本仍然是巨大的。另外，围绕 Linux 开放源代码所产生的知识产权纠纷常常为产品制造商带来意外的麻烦和成本。

不过，如果手机制造商没有能力开发自己的软件 and 用户界面，软件集成商就可以代为完成操作

系统与应用软件的相关工作。于是，风河这样的公司找到了新的市场空间——提供集成服务，并且确保手机 Linux 软件的整体质量，包括筛选和过滤手机软件中所用到的所有软件代码，不仅确保其质量与安全，而且确保不会无意间侵犯到别人的知识产权。

相对于分析师们指出软件在手机原物料成本中所占比例达到 25%，风河公司移动产品部门总经理 Jason Whitmire 认为实际的比例更高。他说：“软件在手机中所占比例的上升速度出人意料，目前大约在总成本中所占比例大约为 30%，而在 12 个月内将会增加到 40%。免费且开放的源代码具有非常大的诱惑力。手机厂商的软件成本因此而降低，并且更容易找到熟练的开发工程师，甚至让移动通信运营商更加方便地定制自己的软件功能。但是，实际上处理软件问题比想象的更为复杂。你需要把许多东西组合起来，并且确保它们可以顺畅地协同工作。如果你是运营商，你还需要自己特有的软件版本。我们已经在跟运营商携手配合，提供他们专属的 Android 软件架构，支持他们实现自己特许的业务功能。这样，运营商就可以反过来把这些特许的业务授权给他们的 OEM 厂商。”

风河不仅同时跟 LiMo 和 OHA 的成员厂商保持着密切联系，而且与 Intel 紧密配合，推出了专门针对 Moorestown 芯片组优化的 MID(mobile Internet devices, 移动上网设备)软件平台。Whitmire 相信，在今后 5 年内，风河公司面向 LiMo、OHA 和 Intel 的三种 Linux 组合将会在 Linux 移动设备中大行其道。

谁将最终胜出？

在手机软件平台中，开放的 Linux、半开放的 Symbian、属于一家公司的 Windows Mobile 乃至苹果、黑莓和 Palm 谁将胜出？这似乎是开放平台与私有平台之间的大对决。

对于软件开发者来说，Windows Mobile、苹果、黑莓和 Palm 最大的吸引力就是平台的单一化。只要是针对同一个平台，他们不用担心自己开发的软件在这支手机上很好用，到了另一支手机上可能就不能用了。

Linux 平台则不同。首先，全球有超过 900 家运营商在 200 多个地域运营移动通信网络，每家运营商都存在特殊的需求，他们都需要对 Linux 原始内核进行变动，从而影响了原始代码的纯正度。另外，存在着数十个 OEM 厂家、数百种芯片变体、数千种手机原型，每种原型都对 Linux 内核产生了各自不同的需求。因此，Linux 手机软件厂商不得不谨小慎微地移植每一个细节和参数。

业界普遍认为，与 Linux 相比，Symbian 是比较成熟的，其集成化已经做得很好。以此为软件平台，可以比较安心。相比之下，Linux 由于参与者众多，而且非常自由，就形成了鱼龙混杂、支离破碎的局面。比如摩托罗拉就在 Linux 手机方面饱经风霜。在 2007 年，摩托罗拉基于 Linux 手机的发货量最高将达到 1000 万部，在其发货的手机总量中占 60%。但是，摩托罗拉基本上是自己维护和集成了内部专用的 Linux 平台，为此所花的费用太过高昂，估计在 5 亿到 10 亿美元之间。整个手机业界都从中学到了教训，于是有了 LiMo 和 OHA 这些推进 Linux 手机开放和标准化的组织。

风河公司的 Whitmire 认为：“Symbian 已经走入生命周期的末期。从好的方面来看，Symbian 已经非常成熟，不过它是由 Nokia 一家公司掌控的，在市场中的整体占有率已经开始下滑。我所接触的手机厂商都没有把 Symbian 列入下一代手机的计划。他们不是选择 LiMo 就是选择 Android。”

所以，市场分析师看好风河在 Linux 手机软件市场的前景就很自然。正如一位分析师所说：“在软件集成领域，风河公司久负盛名。他们最善于把多种多样的好东西塞进小小的百宝囊之中。”

可见，对于走上手机 Linux 之路的厂商和开发者来说，如果有风河这样的帮手，应该不至于步履蹒跚吧！

技术沙龙

沙龙服务社区 交流走向成熟

---ChinaUnix 北京上海两地技术沙龙掠影

过去一个月中，ChinaUnix 技术沙龙先后连续在北京和上海举办两场，为 ChinaUnix 春季技术沙龙划上了一个圆满的句号。

北京站运维监控论坛掠影

2009 年 3 月 28 日下午，在北邮网络学院会议厅，运维监控沙龙顺利结束了。本次技术沙龙参与人数近 180 人。大家非常积极，原本 13:00 签到，13:30 正式开始，实际上不少人 12:00 已经赶到现场，而 13:00 已经坐满，中途加了两次椅子，直到下午六点多时还基本全场满座。这次北京名为“运维监控”技术沙龙，云集了来自各行各业的企业运维工程师。相比较 09 年一月份 IT168 年会的那次 CU 网络优化论坛，超过一半的人来自于互联网类企业，而本次沙龙则行业分布较平均，有不少用户是来自传统行业，比如教育、制造、物流、电信、保险等。虽然这次活动在北京举行，却吸引了不少外地的企业和技术人员前来参与，比如有专程从杭州过来的淘宝运维工程师、郑州过来的高校运维经理、来自上海的监控行业专家，以及天津、济南等地的企业运维工程师。

同样，在总结以往沙龙活动举办的经验后，我们在本次活动中加入了一些创意，比如更加贴近工程师视角的现场技术演示。为此，我们专门邀请了 CU 论坛著名服务器监控问题专家田逸先生，花两个多小时来给我们现场演示并讲解，如何用 Nagios 监控关键业务网络。虽然这次创新在时间控制方面出现了一些小问题，但是总的来说，形式和效果都得到了现场参与人员的一致好评。从整体上看，本次技术沙龙是比较成功的。其后，曹世军、曹金成，以及来自上海的李伟等演讲人员，都从其他几个方面，和用户进行了精彩的技术交流。

-----北京站活动部分照片-----



这是本次技术沙龙的场地的全景图，座无虚席，连走道和门口都站满了人，此次沙龙活动的场地是由北京邮电大学网络教育学院提供，在此，非常感谢北京邮电大学网络教育学院网络信息中心的李主任、张主任、史工等。

下面是主题演讲人：分别是田逸、曹世军、曹金成、李伟：



在台上讲的是高手，其实，在台下的参与者们，同样高手云集！例如在以上几位在演讲时，台下不少人会进行补充，或者提出其他的意见。



上海站网络优化高端论坛掠影

这次上海站技术沙龙是我们第一次如此大规模的在华东地区组织的沙龙活动，而且话题非常具有针对性，主要面向企业网络优化相关从业人员。同时也穿插了其他一些相关技术议题，用来保持整个会场的连续性，比如还谈到了 Linux 系统优化、网络负载均衡、数据库容灾备份，以及网络交付架构优化等话题。

在上海同事们的精心组织和相关合作伙伴的全力配合下，本次名为“网络优化高端论坛”的 CU 技术沙龙，于 4 月 11 日下午，在上海市江苏路 500 号的电信世界大厦如期举行。本次技术论坛，云集了来自上海周边地区各行各业的企业运维工程师、DBA 等。上海周边的主要互联网企业基本都到场了，例如，51.com、淘宝、51Job、九城、完美、巨人集团、盛大集团、久游网、百合网、麦客网、九维网、篱笆网、淡香商贸、春秋集团、同程、腾讯、易趣、中华网游戏集团等。另外，还有大量的传统企业到场，例如，上港集团、宇达电脑、昆山飞力达物流、新申实业、天马微制造、汉得、广众、凯杰生物、亚太制造、IDT 制造、百联电子、联华电子、谷嘉实业、美帝费集团、中达电子等等。此外，还有长沙的湖南卫视、广州、南京、青岛的企业参与。

这次活动，得到了上海周边地区企业的大力支持，很多企业都是在同事的推荐、或者运维负责人的带领下，结伴过来，在此，非常感谢大家的地下式宣传，以后，也非常欢迎大家介绍同事过来参与。

-----上海站活动部分照片-----



同样，本次活动非常感谢上海交通大学网络教育学院提供了场地赞助，他们提供了很好的 160 多人的多媒体教室，环境非常好，非常适合技术人员进行技术交流。该教室的现代化多媒体网络教学系统，是我目前为止见到最好的教学会议平台，非常不错。

主题演讲人：分别是匡萃彪(FinalBSD)、王振武、孙嘉彬、沈冠宇、徐景春、王伟：



上海站活动总结：

1、时间控制：

这次活动原计划 13:30 正式开始，但推迟到了 13:45，再加上演讲超时，导致活动 18:30 才结束，让不少人长时间饿着肚子（不少人未吃中午饭），在此表示真诚的歉意，下次我们一定严格控制时间。由于时间关系，中场的休息时间被压缩了，还有几位串插互动环节也被压缩了，以后，将重新增加互动的的时间。

2、主题专一性：保证更多人的互动时间

其实，在以前多次技术论坛中，有不少用户反映一次沙龙有太多的不同主题，因此，每个主题的演讲时间有限，一般在 1 个小时以内，因此每个主题不可能讲得有多少深入。正是基于这种情况，我们在 3 月 28 日的北京站运维监控沙龙中，专门邀请了 CU 论坛著名服务器监控问题专家田逸先生，

花两个多小时来给我们现场演示并讲解。另外，安排其他的演讲者进行同一主题的剖析。

这种针对同一主题的技术沙龙，并保证更多人的互动时间，将是我们的以后主要发展方向。不过，因为本次上海站技术论坛，在上海属于第一次举办，所以内容有所广泛，下次将专一化。

沙龙技术资料和相关总结

北京站资料下载
 田逸：运维监控连载：开源监控利器 nagios 实战全解.rar (423.94 KB)
 曹世军：使用 cacti 搭建企业监控平台.rar (1.45 MB)
 曹金成：mysql 运行监控全解.rar (47.45 KB)
 李伟：实战：Nagios 配置 web 方式管理和性能数据展示.rar (2.39 MB)
上海站资料下载
 徐景春：企业级开源数据库灾备体系.rar (1.96 MB)
 王振武：系统平台优化：XEN 虚拟化的实践.rar (40.31 KB)
 沈冠宇：梭子鱼应用交付解决方案：应对网络流量峰值瓶颈.rar (1.36 MB)
 FinalBSD：通过 HAProxy，构建开源负载均衡架构平台 998.rar (709.61 KB)
更多资料下载：
北京站活动链接： http://linux.chinaunix.net/bbs/thread-1064851-1-1.html
上海站活动链接： http://bbs2.Chinaunix.net/thread-1386836-1-1.html

广州站&深圳站意见征集

应广大华南网友的请求，ChinaUnix 将在下个月（5 月份），发起网络优化论坛广州站或深圳站的技术沙龙活动，目前还有两个问题急需确定，请华南地区网友前往论坛相关讨论和建议区帮助我确定相关事宜。链接如下：

<http://linux.chinaunix.net/bbs/thread-1064851-1-1.html>

目前主要想确定两个问题：

- 1、到底在广州，还是在深圳？
- 2、网络优化论坛的议题，具体锁定在哪个点上？

请大家跟贴发表自己的看法，以便于我们展开工作，谢谢！

专家专栏

MySQL 数据库技能 互联网行业从业必备

-----CU 论坛 MySQL 技术征文一等奖获得者许景春专访

ChinaUnix: 首先介绍下徐景春，CU 论坛上的 ID 是 talen-t。曾就职于腾讯，现任 51.com 网站数据库团队 Leader，专职 DBA 工作 4 年。他的工作经验肯定会对现在即将进入 DBA 行业或者正在学习 MYSQL 技术的新人有所帮助。



ChinaUnix: 这次 MySQL 技术征文获得第一名有什么感受，有什么经验心得与大家分享吗？

徐景春: 首先感谢主办方 ChinaUnix 社区、ITPUB 社区、爱可生、Sun 主办的这次活动，其实参加征文主要是分享我们整个团队在 MySQL 应用上的经验！团队里还有 MySQL 专家谭俊青，CU 论坛的 ID 是 showsa，我们的资深备份工程师周枫，还有我们 Oracle 和 MySQL 都擅长的姚艳庆，以及我们的架构师李杰，等等。我们的主要保障数据安全稳定高效的运行，为大数据量，高并发提供有力的支撑。我个人的工作主要是在团队建设，协调组织上，发挥每个人的优势，共同完成我们对数据库的一些想法。

MySQL 最大的特点是免费，且性能不差，在多个国外的评测中都不太差于商业数据库，例如 Oracle，相比商业数据库，MySQL 小巧，灵活部署，易于维护，基本覆盖了关系数据库的绝大部分功能，根据我们多年对 MySQL 的应用，感觉特别适合是非敏感数据，多 PC Server 的分布式数据部署 MySQL 的一些功能相比商业数据库就有一定的差距，比如分区的很多致命 BUG，基于逻辑上的复制导致数据一致性不是很严谨，Cluster 上的不完善等等。

MySQL 的一些 Web 化的工具例如监控，备份，管理这块是收费的，而且相比商业软件还很不全面，这也给了很多热爱 MySQL 的爱好者进行一定的 DIY，定制适合自己企业的产品工具，例如我们这次的灾备体系的平台就是我们根据 MySQL 的特性自行开发的。

总的来说，MySQL 虽然还有很多不完善的地方，但仍然不失为一个好的数据库，否则也不会全

世界有这么多使用和热爱 MySQL 的企业和用户了。

ChinaUnix: 你是什么时候从零开始让自己成长为一个数据库主管的?

徐景春: 这个原因很多吧, 有机遇也有个人的努力, 但肯定要感谢我在腾讯的 leader 汤建国, 是他把我领入了 MySQL 这个领域。然后在整个 paipai 这个非常优秀的团队, 学到了非常多的技术, 管理, 运营知识。后来我来到的现在的公司, 在公司领导的信任下, 以及公司数据, 压力爆炸式的增长下, 不断获得锻炼, 才有了现在对 MySQL 一些经验沉淀。

ChinaUnix: 你对现在的新人有什么建议?

徐景春: 作为新人我觉得首先要熟悉 LAMP, 然后热爱 MySQL, 对 MySQL 感兴趣, 最重要的是还是有机会进入国内几大互联网公司得到锻炼。因为没有大量的数据和压力情况下经验很难得以增长。

ChinaUnix: 新人学习 MySQL, 就业方面如何定位? DBA 的就业需求如何?

徐景春: 目前来看 MySQL 更深入的研究与应用还是在互联网行业, 比如国外的 google, yahoo, 国内的腾讯, 百度等, 所以想专职 MySQL 研究, 还是就业定位于互联网行业。MySQL 就业来说目前应该说很紧缺, 淘宝, 搜狐等国内大部分互联网公司都急需, 但可惜的是目前国内经验丰富的 MySQL DBA 太少。顺便做个广告, 愿意来上海发展的 DBA 可以联系我, 我也在招人, 呵呵!

ChinaUnix: 你对 MySQL 的前景有什么展望? 针对甲骨文收购 SUN 对 MySQL 带来的影响, 你有什么看法?

徐景春: MySQL 的前景应该说毋庸置疑, 从 Oracle 多次收购 MySQL 未果, 以及 SUN 和 MySQL 联姻, 都看得出业界对 MySQL 的高度认可, 加上 google 这样的公司持续的对 MySQL 进行二次开发, 可见 MySQL 的前景是非常好的。特别是随着现在金融危机的到来, 很多传统行业也开始选择采用 MySQL 来节约成本, 这给 MySQL 进入更多的领域。

对于 Oracle 收购 sun, 对于 MySQL 来说, 我不能判断对 MySQL 是好事还是坏事, 但从我的判断来看, oracle 收购 sun, 看中 2 块, 一块是 MySQL, 丰富 Oracle 的用户层次, 一块是 Solaris, Oracle 一直想自己搞 Oracle linux, 似乎不成熟, 正好弥补。

ChinaUnix: 谢谢您接受我们的采访!

徐景春: 不客气!

网友来信

我们的开源世界需要更 Easy

——中学生使用 Linux 的艰难

在高一的时候，经强大的学长的介绍，我向 Ubuntu 官方订了一张 live-cd。在等那张远洋光碟的时间里，有事没事我都会上网搜索相关的讯息。就这样，我成了鲜有的中学 Linuxer。

那个时候的 Ubuntu 还只是难看的 7.10，对于一个刚开始学编程的孩子来说，从 Windows 转到陌生的 Linux，真是个不小的挑战。第一次安装的时候，在分区那里，那个叫迷茫啊！只得点了自动分配，看系统默认的是怎样的，再回到自选那里调整下……基本系统装好了后，就对着那个传说中的新手入门必看 wiki 文，从头到尾，有用没用，不知所谓的刷了一遍……还好，那系统看起来还安好……之后，我几乎是用了一个月才熟悉了 Ubuntu 的基本操作（把终端当 cmd 用，其它操作图形化界面）而已……为什么上手这么慢呢？要知道，住宿的孩子每个星期就那可怜的周末才可以回家用电脑啊，经不起折腾的……

在之后的学习中，遇到过大大小小的问题，有的转转脑筋就解决了，有的实在不知所措，茫然啊！当时还不知道 irc 这东西，就只好问 Google 大神。但 Google 大神也不是万能的，所以还会问一些在饭否和 twitter 认识的朋友。这些朋友对待我的态度大致可以分成两类，一类呢，把我当成 IT 民工，一个劲的跟我说很多我压根没听过的概念，只好在和他们聊天的时候常备 Wikipedia；另一类呢，代表性的一句话是：你还是小孩子，要特殊考虑的。显然，后者肯定是经常被我烦的……

每每解决一个问题，我都会记录在博客里。写得最好的一遍当属《Powerful KVM》（外链最多~），可惜的是，我原来的域名到期了，那些外链也就都失效了……同时，也正是因为这篇文章，我才有幸为 CU 写篇文章^_^

当时折腾 KVM 的最初目的仅仅是为了用万恶的 M\$ Office，wine 出来的 M\$ Office 实在不敢恭维啊！为什么不用 OOo.org？因为那家伙输出的 ppt，在 M\$ Office 里打开就不是给人看的了……而学校是 M\$ 的世界（羡慕台湾的小学生啊）……但是，KVM + XP + M\$ Office 2007（本人属华丽型的 ppt 制作者~）仍是慢得一塌糊涂啊！无奈之下，只好把 XP 装回去，处理好 grub 用双系统~

可是，自从装回 XP，Ubuntu 的使用频率就低下来了……一个著名的问题就出现了：这些 Windows 都能搞定，为什么要用 Linux？还是迷茫啊！这个高深的问题，对我这种鸟来说，还是避之为妙……

高一就在这样的飘荡中过去了。高二呢，想换个发行版，如 Arch 之类的来折腾，却发现这些个发行版的配置真叫折腾了！本身高二就没高一那么有空了，唉，只得继续混合使用 Ubuntu + XP 了。不过，这个时候就比高一时有觉悟多了。对于能用 Ubuntu 做什么，能用的就用，不能用的就用 XP（这句话怎么这么耳熟……）。

好吧，说了这么多，无非是想说，Windows 转 Linux 难，而当这事发生在中学生身上就更难了！主要表现为以下几点：

1. 时间不多，基本上只有周末有空折腾；
2. 技术能力低，毕竟不是人人都喜欢和终端整天在一起，中学就折腾编程之类的就更没几个了；

3. 资料基本都是以程序员对程序员的口吻来写的，不够通俗；
4. 资料库仍未健全，汉化程度也不高；
5. 更加适用的发行版没有出现。

虽然现 Linux 大军的主力是 80 后（还有 70 后？），但即将冲入大学的 90 后也是一股不容忽视的强力军呀！总的来说，我们需要化繁为简，降低那该死的入门槛儿，发展 Linuxer 得从娃娃抓起^_^

最后说说我的设想啊。我的想法是创建一个针对中学生的发行版（当然，我没那能力呢，不然它早就出现了……），它的主要特点如下：

1. 傻瓜式安装流程，参考 wubi 吧；
2. 默认主题有华丽和清新两种，在安装时给予选择，因为对于浮躁的学生来说，外观是很重要……；
3. 软件上，集成好常用的咯，同时尽量多集成些学习和娱乐用的，这样，安装好基本系统后不怎么需要折腾就可以满足日常使用啦，多好啊~；
4. 减少终端出现的机会……
5. 尽可能便捷的求助方式。

其实就是想说，Linux 的世界需要更 Easy，更深入的折腾等入门了再来嘛^_^（再说，咱们的最高指示不是说要我们“不折腾”么~）

作者简介：

大家好，我是 Leewings。我目前就读于广东省深圳中学，即将高三了.... 现任 2009 届深中凤凰木网络社区管理员及其下属社团社长。

目前个人水平可以用广而不精来说，学习过的语言有 Pascal(极少使用了)，C(有待深入学习)，Python(略懂)，PHP(略懂)，CSS(略懂)，SQL(略懂).....

我的博客(技术+生活)是"逆时(www.BackTimer.com)"，欢迎前来指教。

本期推荐

开源盛世或危机——Sun 被收购后

ChinaUnix 网友: jeanlove

利用黑色一分钟的攻势, Oracle 这位神秘的红衣主教把 Sun 这块肥肉从 IBM 这位蓝衣主教的囊中夺走了。当然, IBM 原本就是企业服务行业的老大, 这次收购短时间内不会改变老大的地位, 只是 Oracle 变大了以后, 冠军的宝座有了强力的竞争对手。

原来的 Oracle 很简单, 只有数据库, 后来围绕着数据库衍生了无数的应用软件, 这是和 IBM 竞争的基础, 少些东西, 没有 IBM 那样的一体化解决方案。IBM 为用户提供了一揽子解决方案, 高低搭配均可--硬件有 IBM S/Z/power, 操作系统有 AIX+SuseLinux, 数据库有 DB2+informix, 应用软件有 Lotus+OpenOffice, 有自己的 JVM 和 J2EE 服务器 websphere, 用户选择 IBM 就等于得到了 turnkey 的承诺。现在 Oracle 也加入了这个行列, 硬件有了 SunSparc 产品线, 操作系统有了 Solaris+Unbreakable Linux, 数据库有了 Oracle+MySQL, 应用软件有了 Oracle dev Studio+Sun 的开源产品系列。于是, IBM 和 Oracle 成为了两家几乎互为克隆的公司, 一如麦当劳和肯德基的关系。

接下来的形式就很明朗了, 开源软件在 IBM 和 Oracle 的地位也更加明显, 那就是越来越被证明是商业软件的一种补充。其实自从 Eris. S. R 和 Stallmen 分道扬镳以后, 开源软件和自由软件的割裂也越来越大。自由软件的出发点是完全替代商业软件甚至做的更好, 就像知道了 Emacs 和 mutt 的威力以后就不会沉迷于某个公司开发的 IDE 或者华丽的 GUI 软件。但是开源软件迄今为止没有一个明确的中心思想, 不管它号称如何的带给企业和客户以各种灵活性。从目前的趋势看起来, 对于开源大力支持的 IBM, Oracle+Sun, 只是在把开源作为一种工具而不是目标, 目标可以是为了和微软更好的竞争以及相互竞争, 或者是为了提供商业软件的低端替代产品, 把用户继续绑定在自家的产品线之上, 或者是为商业软件提供一个项目孵化平台, 条件成熟的时候就选择闭源而停止开源。这种理念和 GNU 的把软件彻底开放的终极目标毫不相干, 蜕变成了完全的手段和过程。

无论我们原先多么的盼望有朝一日开源的 MySQL 数据库能超越 Oracle, 现在已经完全没有了这样的可能了。Foxbase/Foxpro 那样的命运, 是否会降临到 MySQL 身上, 现在仍然是一个未知数。如果 Oracle 下一步继续收购 Redhat 的话, 那么将意味着 Open Source 和 GNU 的彻底决裂, Open Source 将专门为商业服务。Redhat Linux 和 Novell SuSE Linux 将正面对掐, Oracle 将不遗余力的排挤其他开源的数据库(无论哪种许可证制度); 由于某种开源 Web 服务器的产品受到商业公司的占有和保护, 用户的自由在减少。或者由于某个开源产品得到了商业公司的大力发展和推广, 其他同类型的开源产品不可避免的有陷入衰落的危险, 而一旦其竞争产品发展跟不上了脚步, 这个发展壮大的开源产品就非常有可能被拥有它的公司选择停止开源, 从而伤害到整个开源社区。这就陷入了一种悖论, 开源选择和商业合作来更好的发展自己, 而一旦失去了绝对控制权以后, 最终自己得到的很可能是伤害。用租赁自己的自由来获得一些好处, 大概率的发展趋势是整个社区既失去了自由也失去了应得的利益。

那么开源繁荣和热闹的背后在往哪里走? 独立自主的联合开发是 GNU 的精神, 引进和合作也是开源的精神, 但是似乎前景不同。一种是自主研发, 获得自由, 虽然研发过程艰辛成果也不见得斐然, 一种是引进生产线合作生产, 用别人一部分技术, 自己合作生产和销售。看起来第二种方法

产生效益更快，只是从长远来看，受制于人，可能一直只能在产业链条上的某个环节打工，最后沦为打工的奴隶，或者是干脆被收购和消灭。专利是一种传染性的东西，一个开源软件当中只要包含了一个原子的专利成分，它就是不自由的，并且它的分支和衍生物也会是不自由的。开源的初衷是为了给我们更加灵活的自由度，但是事情的发展超出了预期，因为没有类似 GPL 的自由契约的保护，灵活性随时可以被商业利益取代，开源的发展正在朝商业软件的马前卒驶去。无数的开源项目，冥冥中有一种趋势便是，始于自由，终于商业。

甲骨文收购 Sun 开源阵营很受伤

ChinaUnix 网友: wolfop

甲骨文收购 Sun，总的来说，Sun 旗下相关开源项目和产品线加大大受到影响。

- 1、MySQL 可能被压制，虽然不一定走向 close。一直以来 Oracle 内部的声音就说过，Oracle 不会死在竞争厂家手上，不会死在自己手上，只会死在 OpenSource 上面。而且 Oracle 内部数据库的研发是最强的，根本不会瞧得上 MySQL 的研发团队。不过对 GPL 的 MySQL 没法彻底消灭，那就放慢研发速度，减少新功能引入，引导客户迁移到 XE 或者 ORACLE 标准版本。
- 2、NetBean 不好说，放弃 NetBean 可能性居大。收购了 Sun，三个 IDE(jdevelop/beaworkshop/netbean)弄死 Oracle，怎么也不可能一起发展的。感觉放弃的机会大，因为 jdevelop 已经基本灭了 bea workshop，Oracle 只是迫于压力提供了 eclipse 开发 weblogic/bpel 之类的插件。IDE 又不赚钱，肯定是缩减研发团队。
- 3、btrfs，这个更玄乎。Oracle 研发这个是因为以前没有自己硬件、没有自己的 OS，想靠扶植 Linux，让用户用 X86/X64 服务器去和小型机强市场。如今自己有高端小型机，还扶植 Linux 干嘛，X86/X64 有自己的 Solaris 操作系统和 zfs 文件系统足矣。减缓研发投入，让社区自己去研发的了。好在现在的 ext4 还比较争气，Linux 到也不用太担心这个。
- 4、OpenSolaris，个人感觉会维持研发投入，Larry 强调了这次收购最重要就是 Solaris 和 Java。但是肯定让 License 没法和 GPL 兼容，让 Linux 没法采用其中源码，打击对手。
- 5、GlassFish。这个还会投入多大力量研发？应该不太可能了，自己的 oc4j 和 weblogic 都头大呢，再来个 GlassFish？
- 6、总体来说，Oracle 对开源的态度不如 Sun，不如 IBM。想想看，Oracle 毕竟一直以来靠软件 license 存在，不是主要靠硬件和服务，而且 Larry 又是一个说一不二的人，对 opensource 来说，不见得是什么好事。

Sun 搞砸 MySQL 不要责怪甲骨文

甲骨文日前宣布以 74 亿美元收购 Sun，与此同时，2009 年 MySQL Conference & Expo 大会也拉开了帷幕，随机开发者已经开始在微博客 Twitter 上表达他们对该交易的看法。其中一名开发者在 Twitter 上表示，“施瓦兹不会在 MySQL 大会上用午餐了。我想他正在前往塔希提岛。”

我认为，这条消息非常好的点出了 Sun 首席执行官乔纳森·施瓦兹(Jonathan Schwartz)与 MySQL 之间 16 个月的关系。自从在 2008 年 1 月收购 MySQL 以来，施瓦兹一直在继续发展 MySQL，尽管在这段时间经历了灾难性 5.1 版的发布，以及关键项目开发者的离去，其中包括 MySQL 联合创始人 Monty Widenius 和 David Axmark 以及 MySQL CEO Marten Mickos 的辞职。

现在，全球最大商用专有数据库厂商将拥有全球最大的开源数据库项目，等待不安的 MySQL 的命运将会是什么呢？

MySQL 将获得新的发展

乐观者认为甲骨文看到了 MySQL 的真正机遇。前 MySQL 首席执行官 Mickos 就是持这种观点的人之一，他在 Twitter 写到：“我认为甲骨文不会扼杀这个业务。拉里·埃里森非常聪明。我离开的时候 MySQL 的每天下载次数大约为 7 万次。它对年轻开发者具有令人惊讶的吸引力。保留 MySQL 对甲骨文有益无害。”

Mickos 认为，甲骨文将借助于 MySQL 来改进其专有数据库产品，从而在低端市场上更好的与微软竞争。尽管如此，所有这些下载不一定会转化为收入，Mickos 和其它人都明白这一点。不过 Mickos 的观点有一定道理。MySQL 深受年轻开发者的支持且拥有巨大的安装群体。甲骨文不会轻易扼杀它的发展。当然并非所有的人都和 Mickos 那样乐观，实际上很多人认为，甲骨文会重新翻版其对 Siebel 的收购；停止被收购者的运营，从而更好的发展甲骨文自身的产品和服务。

MySQL 生命将终结于甲骨文

反方阵营认为，甲骨文收购 MySQL 的目的就是消灭它。

当然，甲骨文不会毫无掩饰的取消该项目。更可能的情况是，甲骨文将继续过去一年中 Sun 所提出的策略：任由这个项目自然发展，而不投入领导者和资源来合理的开发和管理他。几乎可以断定，这将是甲骨文将选择的方向。原因何在？即使甲骨文也希望积极推动 MySQL 的开发和发行，它们却从收购 MySQL 的第一天开始就面临一个问题：没有人来替它们完成这项工作。

Monty Widenius 在其博客中表示，此前一段时间内，多数 MySQL 项目领导者和管理团队都已相继离开 Sun，剩下的人也已经准备离去。即使甲骨文真心想发展 MySQL，鉴于过去它对开源的举动，它也将面临严重的信任问题。Widenius 继续在博客中写道：由于甲骨文过去在开源界的声誉不佳，它将很难让 MySQL 开发人员留在甲骨文，或者说很难把他们留在这个开源数据库项目中。另外，甲骨文将很难向 MySQL 客户、社区和用户保证，将继续保持 MySQL 的免费和开源。

甲骨文并非完全与开源行业交恶，但是除了令人怀疑的甲骨文企业版 Linux 之外，该公司肯定不是一个开源支持者。我们已经很久没有听到来自 Berkeley DB 的消息。

不要责怪甲骨文

我们知道，MySQL 在此之前实际上已经“死亡”。MySQL 这个史上最成功的数据库未来将遭遇的命运，很容易让我们把矛头指向甲骨文。但是，实际上是 Sun 搞砸了 MySQL 的收购，在其接手 MySQL 的 16 个月中把这个开源数据库弄的一团糟，然后把烂摊子甩给了甲骨文。

不过，甲骨文是否决定担负其修复 MySQL 命运的重担，从而让这个开源数据库继续存在下去，还有待时间来给出答案。

分析称甲骨文收购 Sun 对开源事业有利有弊

据国外媒体报道，甲骨文突然宣布以 74 亿美元的价格收购 Sun，这显然会成为接下来数周内的新闻热点，同时这笔交易究竟会对开源产生何种影响？

对于开源事业来说，有一点好处是非常明显的，即在兼并 Sun 之后，甲骨文将成为一个崭新的、规模更大的开源公司。如果甲骨文不将 Sun 的开源技术应用到甲骨文的市场上，那么这简直就是浪

费金钱。因此，通过这一交易，开源技术有可能成为比 Sun 更大的赢家。

关于这一交易的新闻公告提到了“Sun 旗下两个关键的软件财产”，即 Java 和 Solaris。由于 Sun 长期以来一直大力开发 Java，因此该软件的继续发展相对来说不成问题，甲骨文肯定会继续开发并推广这一软件。甲骨文对 Java 的支持必将使该软件在企业设置领域变得更加重要，并且能够以意想不到的方式促进整个开源事业的发展。

此前曾出现过多种关于 Java 的替代软件。随着这笔交易的发生，甲骨文必定会在 Sun 的基础上更加大力地推广 Java，因此这可以成为该软件的一个复兴机会。这不仅对于 Java 来说是一个好消息，而且对于所有依赖于 Java 的开源项目来说都是好消息，尤其是谷歌的 Android 平台。随着更多技术人员参与 Java 的编程，企业接纳 Android 的速度可能会更快，因此 Android 的编程环境突然间与企业有了更多联系，这是 iPhone 与不能比拟的。

另外，这还有利于 Android 其它方面的发展，例如上网本和其它消费产品等。由于 Java 的复兴，Android 将变得更加受欢迎，因此其创新、研发和推广速度将更快。当然，对于其它基于 Linux 的移动平台和 Symbian 及微软来说，这显然是个坏消息。

新闻公告中提到的另一个关键的软件财产就是 Solaris。Sun 旗下的 Solaris 操作系统一直是甲骨文数据库的主要平台。收购 Sun 之后，甲骨文可以对借此对甲骨文数据库进行优化，采纳使用 Solaris 系统中一些最独特、最高端的技术。对于 Solaris 来说，它与 Linux 竞争中最大的优势在于 Solaris 一直是以一种高端系统的标准进行设计。可以肯定的是，尽管 Linux 一直在试图赶超，但是 Solaris 现在又拥有了一种更大的优势，那就是甲骨文的全力支持。依赖 Solaris 运行的甲骨文数据库必将成为公司业务的重中之重，而这会对开源事业产生两个非常明显的不良影响。

第一个不利影响就在于 Linux 的发展。尽管甲骨文在新闻公告中指出：“甲骨文仍将一如既往地支持 Linux 和其它开源平台，并将继续巩固和强化业内的伙伴关系。”但是这一说法在可信度令人怀疑。甲骨文的 Linux 产品 Unbreakable 一直饱受争议，因为人们怀疑甲骨文到底是否真的支持 Linux 平台，或者仅仅提供用户所提出的支持。现在，随着甲骨文收购 Sun，很可能基于 Linux 的产品 Unbreakable 将不再受到重视。

第二个不利影响就在于 MySQL。甲骨文在收购 Sun 于旗下的同时也就意味着它随之拥有了 MySQL，即一种基于 Linux 的数据库。甲骨文显然不可能大力支持 MySQL，最多也就是提供一种低端的数据库，提供给那些抱怨甲骨文数据库过于昂贵的用户。这样就出现了两极分化的情况：基于 Solaris 的甲骨文数据库将向高端发展，而基于 Linux 的 MySQL 数据库将向低端发展。历史经验告诉我们，这并非万全之策：低成本的产品往往会不断抢夺主打产品的市场份额。

有分析师认为，Sun 之所以具有吸引力，就是因为甲骨文收购 Sun 可以同时打击 Linux 和 MySQL 这两个开源威胁，一个是直接打击，另一个是间接打击，甲骨文试图通过此举将即将兴起的竞争对手扼杀于摇篮之中。这笔交易对于 Linux 和 MySQL 来说非常不利，对于后者来说尤为如此。MySQL 甚至可能会因此而失去继续发展的动力。

另外，也有分析师考虑到，在这场交易中，到底谁是赢家，而谁是输家。

有人认为，IBM 是首当其冲的显而易见的输家。IBM 与 Sun 之间长期而艰苦的谈判最终就像微软与雅虎之间的谈判一样无果而终。就在此时，甲骨文的突然杀出令 IBM 非常尴尬，显得极为迟缓和不明智。对于 IBM 来说，甲骨文和 Sun 的结合显然不是什么好消息，因为二者的结合将极大增强甲骨文在企业客户领域的竞争力，并且横空出世了一个足以与蓝色巨人相抗衡的强大对手。

另外，甲骨文与 Sun 交易产生的一个很重要但是出乎意料的赢家将是 OpenOffice.org。在此

之前，甲骨文从未在桌面办公软件领域推出过具有影响力的产品，而 Sun 旗下的 OpenOffice.org 办公软件将弥补这一缺憾。可以想象，一旦甲骨文开始向其企业用户提供 OpenOffice.org 桌面办公软件，那么 OpenOffice.org 的市场份额将会迅速增加。显然，这将是甲骨文策略：使用我们的数据库，我们为您提供相应的拥有甲骨文全面支持的授权 StarOffice 软件（即商业版 OpenOffice.org）。此举将极大推动 ODF 标准的办公软件和 OpenOffice.org 软件。同时，对于正遭受操作系统市场份额不断被蚕食的微软来说，这也是对一个巨大打击。

当然，哪个开源产品将受益于此交易要依赖于甲骨文如何融合 Sun 以及 Sun 旗下的产品。不过有一点可以肯定，即不管这笔交易给开源带来多少坏处，但这毕竟使可以开源事业吸引更多关注。

甲骨文收购 Sun：带给开源界一缕曙光

时间追溯到 4 月 20 日晚，IBM 收购 Sun 终于告一段落，究竟花落谁家也已见分晓。甲骨文 (Oracle) 在众目睽睽之下，令众人大吃一惊的将 Sun 囊如怀中，收购价格是 74 亿美金。甲骨文的这一举动，进一步巩固公司在数据库市场的地位，又扩充其在硬件领域的市场份额，可谓一箭双雕。

甲骨文 CEO 埃里森 (Larry Ellison) 说，“我们收购 Sun 将改变 IT 业，整合一流的企业软件和关键任务计算系统。甲骨文将成为业界唯一一家提供综合系统的厂商，系统的性能、可靠性和安全性将有所提高，而价格将会下滑。”

事件之大，引来各界人士纷纷议论。醉翁之意不在酒，埃里森表示，基于 unix 的操作系统 OpenSolaris 是其看重的原因之一。长期以来，Solaris 一直是甲骨文数据库运行的主要平台，用户群体相当庞大。这样一来，在今后的日子里，甲骨文一定会大力推广基于 Solaris 的 oracle 解决方案，而且 Solaris 系统的性能和速度都令众人满意。众人的口碑相传，加上厂商的大力推广，软件+操作系统的完美结合，甲骨文的势力将锐不可当！

甲骨文同时还是 Linux 的拥护者，因此权衡两种操作系统之间的关系颇受外界瞩目。曾经甲骨文首席架构师爱德华·斯格列文表示，甲骨文将把 Linux 作为其数据中心的首选操作系统，因此客户也不要再选择其他操作系统了。但是 Linux 会取代 OpenSolaris 么？答案不想而知，不会！因为 Solaris 能够带来相当稳定的维护业务营收以及服务营收，因此这么一大块美味的蛋糕怎么会弃而舍之呢？

据某市场研究公司分析师称，整合 OpenSolaris 和 Linux 的最大障碍是两者间许可协议。OpenSolaris 采用了 Sun 的“通用开发和发行许可” (CDDL)，Linux 则采用 GNU 的“通用公共许可” (GPL)。甲骨文可能改变 OpenSolaris 的许可协议，使 OpenSolaris 和 Linux 可以无缝连接。因此也对 Linux 的发展起到一定的推动作用。

此次收购，轰动了整个 IT 业界。操作系统+软件+中间件齐头并进的甲骨文，有谁能够与之媲美？微软是一个家主打“操作系统”的公司，一个 windows 就让微软在业界呼风唤雨；甲骨文是一家软件公司，主打“数据库”，任何公司，任何个人，数据都是最重要的信息，而 oracle 这这方面在业界堪称一霸，加之 Solaris 操作系统的整合，一定会更加强示。

不管将来会怎样，起码现在有个良好的开头。俗话说：好的开始是成功的一半！期待甲骨文的下一个奇迹，希望叫人拍案叫绝！更希望带给开源界一缕曙光！

技术新知

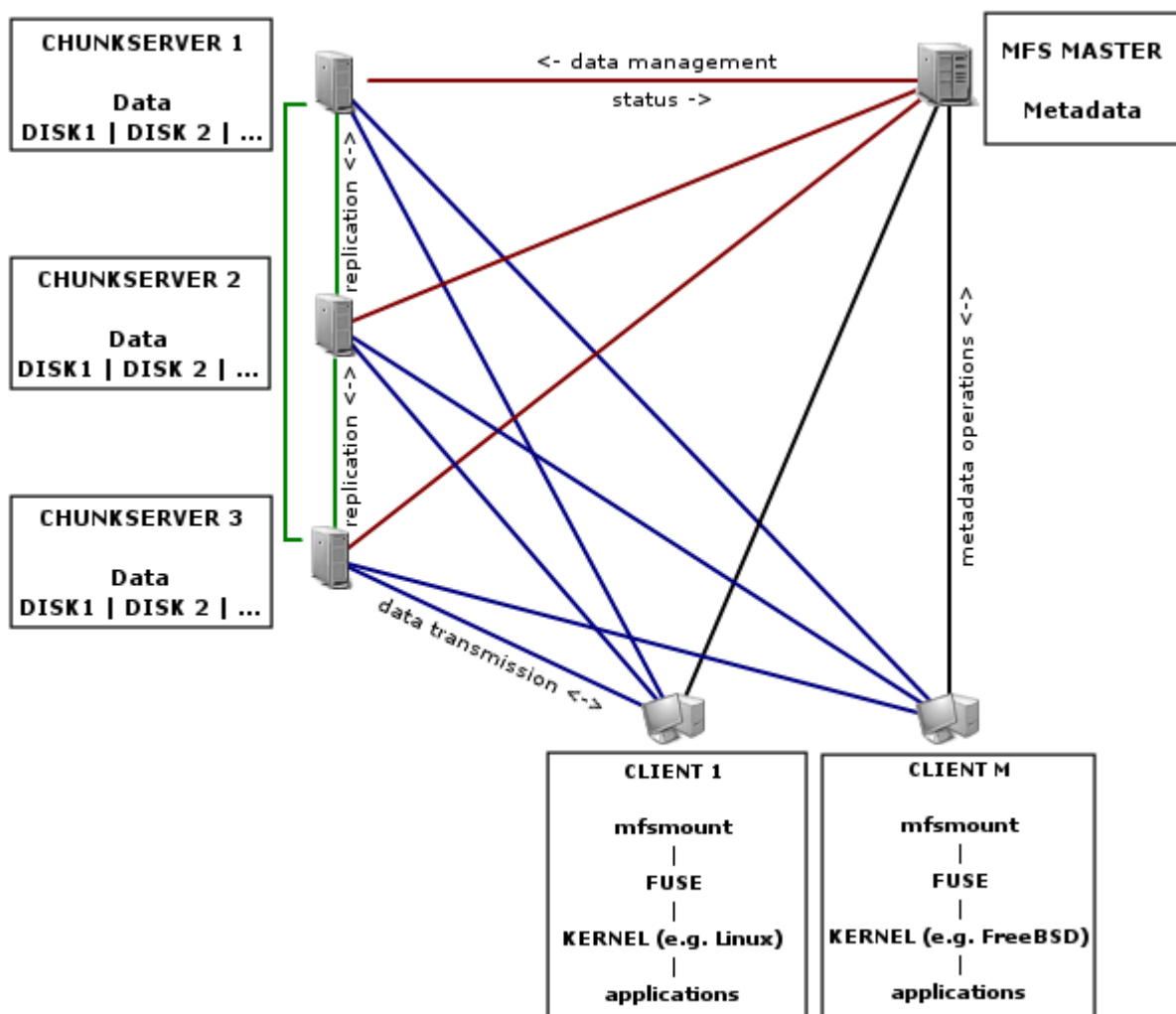
MFS 分布式文件系统架设笔记

ChinaUnix 网友: zhengwei_zw

什么是 MFS 文件系统?

Moose File System 是一个具备容错功能的网络分布式文件系统, 它将数据分布在网络中的不同服务器上, MooseFS 通过 FUSE 使之看起来就是一个 Unix 的文件系统。

MFS 文件系统个人理解: 将分布在各个范围的计算机, 将他们未使用的分区统一进行管理使用的一种文件系统。



MFS 文件系统结构:

包含 3 种角色:

管理服务器 managing server (master)

数据存储服务器 data servers (chunkservers)

客户机挂载使用 client computers

个人理解 3 种角色作用:

管理服务器:负责各个数据存储服务器的管理,文件读写调度,文件空间回收以及恢复.多节点拷贝

数据存储服务器:负责连接管理服务器,听从管理服务器调度,提供存储空间。

客户端:挂载远程管理服务器上所管理的数据存储服务器,通过 fuse 内核接口.看起来共享的文件系统和本地 unix 文件系统使用一样的效果。

架设过程:

服务器 2 台

(最少是 2 台,一台服务器做管理服务器,数据存储服务器,以及客户端 3 种角色,另外一台计算机做数据存储和客户端)

称第 1 台计算机为 A 机,机器上跑 3 种角色,第 2 台计算机称 B 机,跑两种角色.要是你计算机多的话可以只跑 1 种客户端角色或者数据存储角色,根据具体情况决定.

操作系统:FreeBSD6.x or Freebsd 7.x minni 安装 升级 ports 树

1.在 A 机器上安装

CODE:

```
/sysutils/fusefs-kmod
```

```
./devel/pkg-config
```

这两个 ports 包

pkg_info 的结果为

CODE:

```
b# pkg_info
```

```
fusefs-kmod-0.3.9.p1.20080208 Kernel module for fuse
```

```
fusefs-libs-2.7.2_1 FUSE allows filesystem implementation in userspace
```

```
libiconv-1.9.2_2 A character set conversion library
```

```
pkg-config-0.21 A utility to retrieve information about installed libraries
```

```
b#
```

然后在 rc.conf 中添加

CODE:

```
fusefs_enable="YES"
```

2.在 <http://www.moosefs.com/index.html> 下载 mfs-1.5.12.tar.gz

CODE:

```
#fetch [url]http://www.moosefs.com/files/mfs-1.5.12.tar.gz[/url]
```



```
#tar xvzf mfs-1.5.12.tar.gz
#cd mfs-1.5.12
#./configure (这样制作出来的 bin 文件和 sbin 文件以及元数据在/usr/local/sbin 和/usr/local/bin,
配置文件在/usr/local/etc 元数据存放在/usr/local/var/mfs 下) 默认不带任何参数编译出来的,可
以做数据和管理服务器.
#make && make install 就完成了两种角色服务器的安装
```

#####下面是在 A 机上编译出客户机的执行文件#####

CODE:

```
#make clean && ./configure --prefix=/client --enable-mfsmount && make && make install
```

即可,在/client 目录下就产生了一个 mfs 的客户端连接软件

2.在 B 机器上安装

CODE:

```
/sysutils/fusefs-kmod
./devel/pkg-config
```

这两个 ports 包

pkg_info 的结果为

CODE:

```
b# pkg_info
fusefs-kmod-0.3.9.p1.20080208 Kernel module for fuse
fusefs-libs-2.7.2_1 FUSE allows filesystem implementation in userspace
libiconv-1.9.2_2 A character set conversion library
pkg-config-0.21 A utility to retrieve information about installed libraries
b#
```

然后在 rc.conf 中添加

CODE:

```
fusefs_enable="YES"
```

2.在 <http://www.moosefs.com/index.html> 下载 mfs-1.5.12.tar.gz

CODE:

```
#fetch [url]http://www.moosefs.com/files/mfs-1.5.12.tar.gz[/url]
#tar xvzf mfs-1.5.12.tar.gz
#cd mfs-1.5.12
```

#./configure --disable-mfsmaster --enable-mfsmount(这样制作出来的二进制文件可以做数据存储服务器和有了 mfsmount 文件,默认编译是没有 mfsmount 的).

#make && make install 就完成了数据存储角色服务器和客户端的安装

下面开始配置

A 机器上挂载一个空闲的,比较大的分区

CODE:

```
#mount /dev/ad1s1 /mnt/mfs
#cd /usr/local/etc
#ls
-rw-r--r-- 1 root wheel 434 Mar 31 11:34 mfschunkserver.cfg
-rw-r--r-- 1 root wheel 36 Mar 31 11:34 mfshdd.cfg
-rw-r--r-- 1 root wheel 425 Mar 31 11:34 mfsmaster.cfg
```

解释这 3 个文件的作用

mfsmaster.cfg 是管理服务器初始化文件,不需要任何修改既可使用.里面主要就是设置监听端口这些.

我们用默认既可

mfschunkserver.cfg 是数据存储文件配置文件.当数据存储和管理服务器不在一起的时候,要修改

CODE:

```
# MASTER_HOST = mfsmaster 这个值为管理服务器的 ip 地址或主机名
# MASTER_PORT = 9420
mfshdd.cfg 这个是在 mfschunkserver.cfg 中制定的文件名,这个文件里面存放本地存储分区路径.
我这里 mfshdd.cfg 内容就是一行
#cat mfshdd.cfg
/mnt/mfs
#
```

启动顺序

启动管理服务器 ----->启动数据存储服务器---->挂接管理服务器 mfs 文件系统

CODE:

```
#!/usr/local/sbin/mfsmaster start
```

CODE:

```
#netstat -na | grep 942 看是否有两个 942* 为 listen 的端口,若有就开启成功了.
#!/usr/local/sbin/mfschunkserver start
#netstat -na | grep 942 查看是不是多了一个 9422 的端口,若有则开启成功了
也可以使用 sockstat -4 查看
```

如果启动不成功,一般情况就是挂载目录权限问题.默认编译的 mfs 是按 nobody 权限来的.要修改/mnt/mfs 权限为 nobody 既可.

经过上面两部管理服务器和存储服务器就启动起来了.下面是进行本地挂载和在 B 机器上挂载在 A 机器上 执行

CODE:

```
#/client/bin/mfsmount -h A 机 ip -w /mfs  将管理机 ip 挂接到/mfs 目录
```

A 机器上就可以使用 mfs 系统了,挂接点是/mfs

下面是加入 B 机的数据存储和 B 机自己使用 mfs

B 机

#配置挂接空闲的,比较大的分区,然后修改 mfshdd.cfg 内容为挂接地点.修改 mfschunkserver.cfg 的 MASTER_HOST =值

启动 mfschunkserver

CODE:

```
#/usr/local/sbin/mfschunkserver start
```

挂接管理机的 mfs 文件系统

CODE:

```
#/client/bin/mfsmount -h A 机 ip -w /mfs
```

我在做 MFS 试验的时候, 田逸给了我些帮助, 在这里感谢他。

用 vmware 做 mfs 数据存储服务器的时候, 虚拟的硬盘一定要大点, 最好大于 256M 的硬盘。否则 used 永远都是 100%。

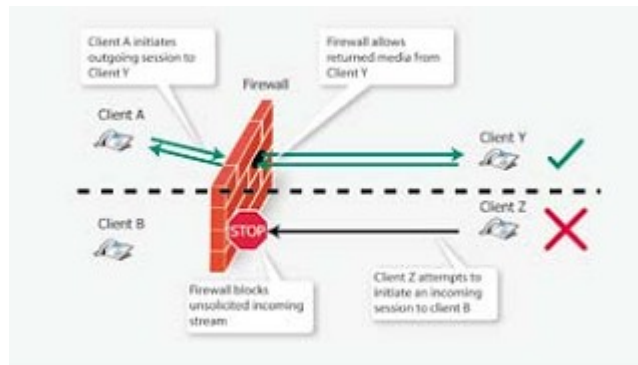
P2P VPN 的今生前世

ChinaUnix 网友: alife

企业发展到一定规模的时候大都在全国乃至全球设置分支机构, 它们的地理位置分散, 在 VPN 没有出现之前, 这些分支机构之间的网络互联只能通过租用专线实现, 然而租用专线的费用高昂, 一般企业无法承受。

随着互联网的发展, 互联网的接入越来越廉价, 嗅觉敏锐的厂家推出了替代专线的解决方案: 在 Internet 的基础上实现一张虚拟的专用网络, 这就是 VPN (virtual private network 虚拟专用网) 的由来。它一方面采用了 Internet 作为传输媒介, 租用成本大大降低; 另一方面通过 IPsec 等多种安全协议实现了等同于专线般安全的效果, 因此普一推出便大受欢迎, checkpoint、cisco、juniper 等都有成熟的商业解决方案, 然而它们针对的是企业市场, 对于个人用户而言还是过于高贵了, 直到 2003 年情况才发生改变。

2003 年，日本筑波大学的一名学生登游大发布了 SoftEther 这款软件，该软件允许用户将两台位于 NAT/防火墙之后的电脑实现互联，这在当时引起了很大的轰动，为什么会这样呢？请看下图：



熟悉 TCP/IP 的读者应该知道，在互联网上通信的双方都必须有公网 IP，然而公网 IP 的数量有限，往往是多个人通过 NAT 方式共享一个公网 IP，所以只能是由私网的用户去访问公网的服务器，无法由公网的用户直接访问私网的主机。这在早期的互联网应用中倒也相安无事，因为当时上网主要是浏览新闻。

随着互联网的发展，p2p 应用层出不穷，1999 年诞生的 napster 顿时名动天下，因为它是第一个以 p2p 方式共享文件的软件。后来相继诞生了许许多多的 p2p 应用，然而它们都只是在应用层实现了 p2p，软件提供什么应用，用户就只能使用这些应用，而 SoftEther 不同，它直接在网络层实现了 p2p，换句话说就是用户透过 SoftEther 可以实现联网游戏、文件共享、远程访问、视频会议等大量的互联网应用，所以这在当时赢得大量网友的青睐，为什么？因为 CS、因为魔兽，这些网络游戏需要在联网对战，而玩友电脑大都隐匿在 NAT 之后，SoftEther 为他们实现了互联的美好愿望。

2004 年 4 月份，SoftEther 公司成立。

2004 年 8 月，SoftEther 正式推出 1.0 版本，开始闭源，开始收钱。

2006 年，推出 2.0 版，更名为 PacketiX VPN，号称重写了全部源代码，跟 SoftEther 1.0 一点关系都没有，与个人用户渐行渐远……

提起 SoftEther 就不得不提 VNN，VNN 是国人开发的一款类似软件，虽然其推出的时间要比 SoftEther 早些，但是一直没有得到广泛的关注，后来 2004 年 5 月份，登游大抱怨 VNN 主页抄袭了 SoftEther 的主页样式，还在日本媒体中大势宣传。当时的 CCF 精品技术论坛上还掀起了一番激烈的讨论 (<http://bbs.et8.net/bbs/printthread.php?t=527295&pp=100>)，这时 VNN 才逐渐进入众人的视野。其实两款软件实现的原理有很大的不同，SoftEther 通过在 ssl over tcp 来保证安全，而 VNN 使用 udp 进行封装；SoftEther 客户端需要在互联网上找一个公用的虚拟 hub 来实现互联，而 VNN 的客户端需要到 VNN 公司的服务器上进行注册才能实现连接；SoftEther 客户端之间的通信需要通过虚拟 hub 中转，而 VNN 客户端注册完毕之后就可以实现点对点的互联。它们之间唯一的共同点就是都使用了虚拟网卡。令人惊奇的是两家发生矛盾的原因竟然是主页过于雷同而不是源代码的抄袭，这在软件史上倒是头一遭，权当作是茶余饭后的谈资。

VNN 是一款闭源软件，不提供服务器版本，用户需要登录到官方的服务器上进行验证之后然后才能建立 p2p 通信隧道。现在由北京宇华亿欣科技有限公司推广运营，已经发展到了 4.0 版，功能很丰富，只是需要收取服务费，这表明它已经不跟个人用户玩了。

有压迫的地方就有起义，有需求的地方就有市场，2005 年，LogMeIn 公司看中了 p2p VPN 的

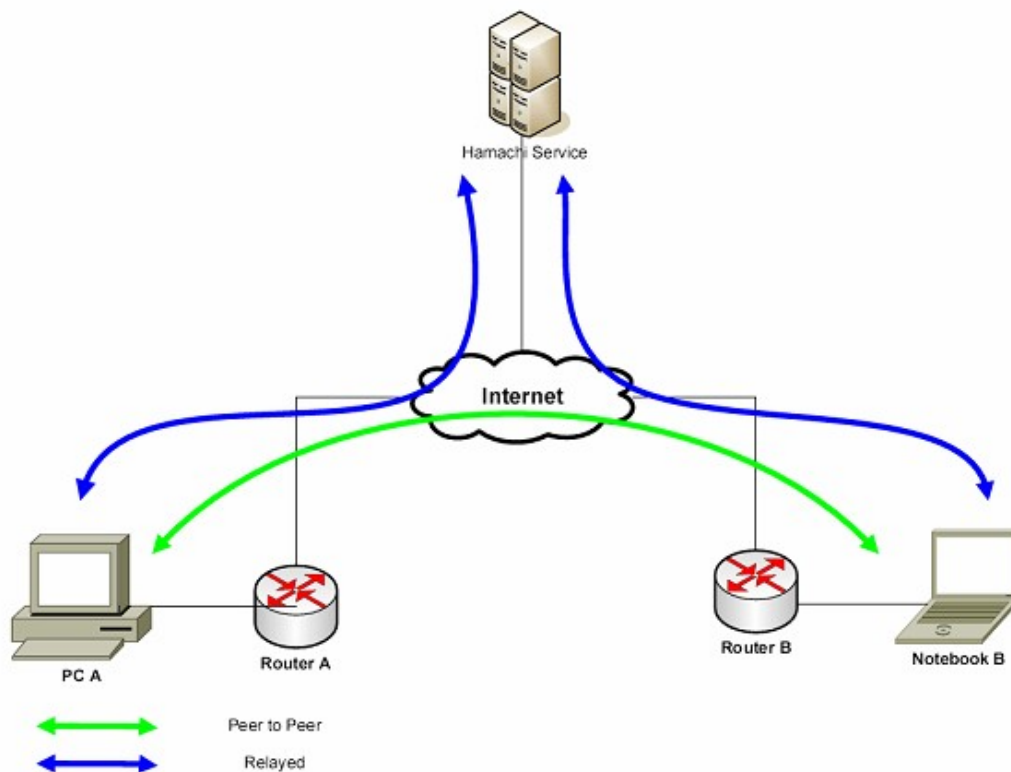
个人市场，推出了 hamachi，它的实现原理跟 VNN 类似，而且可在多种 OS 中运行，win2k、win2k3、vista 自然不在话下，还可以在 Linux 和 OSX 下跑，甚至破天荒的推出了 Nokia770 的版本 (<http://files.hamachi.cc/linux/nokia-770/>)，这对移动商务人士来说可是一大噱头。时至今日，hamachi 已经发展到了 1.0.3，值得一提的是 hamachi 的版本升级非常谨慎，从 0.9.9.9 到 1.0，总共发布了 61 个测试版本，时间跨度为 2 年，除了 foobar 和 wine，估计无人能敌了。

hamachi 相对于 SoftEther/VNN 的优势在于：

- 对于个人用户免费；
- 零配置，使用非常简单；
- 体贴的管理功能，由用户自己组建网络，管理组内成员；
- 采用了 RSA、DH group、AES256 和 HMAC-SHA1 多种安全技术；

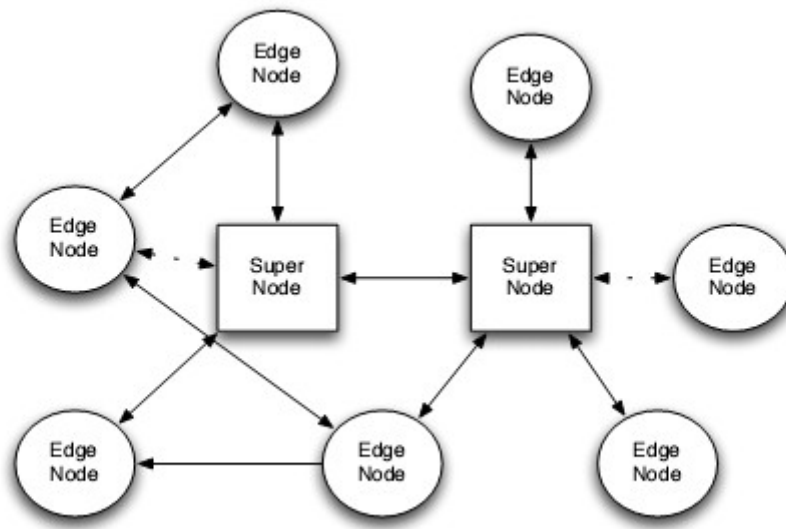
这些优势对个人用户是非常具有杀伤力的，单单免费这一项就可以把 SoftEther 和 VNN 踩在脚下，然而，hamachi 的安全性还是值得怀疑，请看下图：

LogMeIn Hamachi Architecture



用户需要登录到 hamachi 的服务器进行注册才能建立隧道，由于 hamachi 是一款闭源的商业软件，LogMeIn 公司是否会截获用户的数据流就不得而知了，因此，在 p2p VPN 这个肥沃的市场中，软件公司要真的要好好想想商务模式，否则技术实现方案再好，用户不接受也是白搭。

hamachi 就这样不温不火的发展着，直到 2008 年，ntop 的作者 Luca Deri 开始研究 p2p VPN，他一方面看到公众对 p2p VPN 有着强烈的需求，另一方面又不满足已有产品的现状，于是 n2n 诞生了。



如上图所示，n2n 是一个二层架构的 VPN 网络，其中 super node 提供场所，让两个位于 NAT/防火墙之后的 edge node 进行会面，一旦双方完成首次握手，剩下的数据流就发生在两个 edge node 之间，如果有一方的 NAT 属于对称型 (symmetrical)，super node 则还需继续为双方提供数据包的转发；edge node 负责数据流的加解密，原理很简单。

对于一个 VPN 而言，主要涉及封装和解密两个步骤，edge node 使用 UDP 协议进行封装，目的是为了兼容防火策略，因为很多防火墙禁用了非 TCP/UDP 协议。加密算法则采用了 twofish，好处开源、简便，处理速度快。

为了降低设计难度，n2n 利用了 tap/tun 虚拟网卡，这样做得好处是一方面软件尺寸极小，一方面源码的依赖性极低，可以很容易移植到嵌入式设备中，目前有 openwrt 的版本，在未来的计划中，还将移植到 android 和 iPhone 中。

相对于 hamachi，n2n 最大的优势在于：

1. 开源，任何人都可以检查代码，看看是否有猫腻，而 hamachi 是闭源的，LogMeIn 是否会截获密钥不得而知，一旦截获密钥，就可以对流经 hamachi 服务器的数据包进行解码。n2n 的加解密过程由 edge node 实现，只有两端的用户知道协商好的共享密钥，super node 无从知晓。
2. 灵活性，n2n 允许用户在 Internet 上自行创建 super node，也可以利用任何一个公开的 super node。hamachi 用户则必须登录到 LogMeIn 服务器才能创建隧道。

n2n 支持的 OS 也非常广，Linux、FreeBSD、MAC OSX、甚至 windows，不过由于 n2n 只提供源代码，需要用户自行手工编译。

下面就介绍一下如何在 Debian Lenny 中安装和使用 n2n。

1. 在 sources.list 中加入 mentors.debian.org

```
alfie:~# vim /etc/apt/sources.list
```

```
deb-src http://mentors.debian.org/debian/ unstable main contrib non-free
```

```
:wq
```

```
alfie:~# aptitude update
```

2. 下载&编译

```
alfie:~# apt-get source n2n
```

;从 mentors 中下载 n2n 的源代码

```
alfie:~# apt-get install fakeroot build-essential devscripts
```

;安装编译环境所需的组件，debuild 是 devscripts 软件包中的一个脚本，编译的时候需要用到

```
alfie:~# apt-get build-dep n2n
```

;编译 n2n 的过程可能需要一些 lib 的支持，该步骤就是让 debian 判断并下载安装相关的 lib

```
alfie:~# cd /usr/src/n2n_1.2.2~svn3653
```

```
alfie:/usr/src/n2n_1.2.2~svn3653# debuild -us -uc
```

;如果不是 Debian 的维护者，需要打上-us 和-uc 这两个标签，避免编译完后签名

3. 安装

```
alfie:/usr/src/n2n_1.2.2~svn3653# cd ..
```

```
alfie:/usr/src/# dpkg --install n2n_1.2.2~svn3653_i386.deb
```

4. 使用

家里的电脑

```
alfie:~# edge -d n2n0 -c linuxabc -k linuxabc -a 10.1.2.1 -l 88.86.108.50:82
```

公司的电脑

```
ds-server:~# edge -d n2n0 -c linuxabc -k linuxabc -a 10.1.2.2 -l 88.86.108.50:82
```

说明：88.86.108.50 是 <http://www.vpnhosting.cz/index.php/N2n-at-vpnhosting.cz.html> 提供的一个公共 super node。

这样就可以将两台位于 NAT/防火墙之后的电脑互联起来了，是不是特别简单呢？

n2n 是一个很有前途的 p2p VPN 软件，优秀的设计理念将会为它带来大量的用户群，随着时间的流逝，SoftEther，VNN，hamachi 将成为过眼烟云，唯有开源的 n2n 才是永恒。

CPU 负载平衡之一运行队列的 load 计算

ChinaUnix 网友 wxc200

在每个运行队列 struct rq 里，load 代表当前运行队列的负载，同时还有一个 cpu_load[] 这样的数组，在我理解，它是一个分级别的代表当前运行队列负载的“替身”。在多 cpu 调度时，会计算不同的 cpu domain 的负载，根据不同的 index，会选取相应的 cpu_load[] 作为当前运行队列的负载返回。

在每个 tick 处理函数 scheduler_tick() 中，会调用 update_cpu_load()，来更新当前运行队列的负载，便于后面 cpu 平衡调度时选取最忙的 cpu。比如调度函数 find_busiest_group()，它的工作

是: find_busiest_group finds and returns the busiest CPU group within the domain. 它会通过两个函数 source_load()和 target_load()来计算并返回当前运行队列的负载。例如 target_load()是这样的:

```
static unsigned long target_load(int cpu, int type)
{
    struct rq *rq = cpu_rq(cpu);
    unsigned long total = weighted_cpuload(cpu); 返回当前 run queue 的 load

    if (type == 0 || !sched_feat(LB_BIAS))
        return total;

    return max(rq->cpu_load[type-1], total); 返回相应 "替身" 与 运行队列负载的较大者
}
```

这两个函数会有个参数 type,它的目的就是选取相应的 cpu_load[]。

在系统初始化过程中, 会把 cpu_load 默认为 0. 有兴趣的朋友可以参考 kernel/sched.c sched_init()函数, linux 2.6.28 line:8286,片断代码如下:

```
for (j = 0; j < CPU_LOAD_IDX_MAX; j++)
    rq->cpu_load[j] = 0;
```

那么后面这个数组是怎么变化的呢? 它的变化趋势是什么样子的呢?

我觉得 update_cpu_load()还是比较有意思, 来一起分析下。

```
/*
 * Update rq->cpu_load[] statistics. This function is usually called every
 * scheduler tick (TICK_NSEC).
 */
static void update_cpu_load(struct rq *this_rq) //当前运行队列指针作为函数参数
{
    unsigned long this_load = this_rq->load.weight; //当前运行队列的负载值
```



```
int i, scale;
```

```
this_rq->nr_load_updates++;    //代表 load 的更新次数 每一个 tick 都会加 1，真够忙的。
```

```
/* Update our load: */ //CPU_LOAD_IDX_MAX 在运行队列结构体中=5,这儿对 5 个等级的  
cpu_load[]进行更新。
```

```
for (i = 0, scale = 1; i < CPU_LOAD_IDX_MAX; i++, scale += scale) {
```

```
    unsigned long old_load, new_load;
```

```
    /* scale is effectively 1 << i now, and >> i divides by scale */ //请注意，这个 scale 就  
    是 2 的 i 次幂。它
```

的值 分别是 1 2 4 8 16

```
    old_load = this_rq->cpu_load; //当前 cpu_load[]数组里面的值
```

```
    new_load = this_load; //当前运行队列的负载值
```

```
    /*
```

```
    * Round up the averaging division if load is increasing. This
```

```
    * prevents us from getting stuck on 9 if the load is 10, for
```

```
    * example.
```

```
    */ round up: //目的 如果 load 是在增长的，不要把余数别浪费了
```

```
    if (new_load > old_load)
```

```
        new_load += scale-1;
```

```
    this_rq->cpu_load = (old_load*(scale-1) + new_load) >> i; //这个公式，我会下面分  
析。
```

它的意思就是根据当前运行队列的负载以及上次 cpu_load[]的数值，计算出当前 cpu_load[]应该的变化。

```
    }
```

```
}
```

现在分析下这个公式:

假设级别为 i(0~4),运行队列的 load 值 M,计算前的 $cpu_load[i] = m_i$,则每次的计算新的 $cpu_load[i]$ 遵循如下规律:

$$\text{cpu_load}[i] = (M - m_i) / 2^i + m_i$$

$i = 0 \sim 4$ 的情况下, 此公式可扩展为:

$$i=0: M - m_i + m_i$$

$$i=1: (M - m_i) / 2 + m_i$$

$$i=2: (M - m_i) / 4 + m_i$$

$$i=3: (M - m_i) / 8 + m_i$$

$$i=4: (M - m_i) / 16 + m_i$$

由此可见, 在 M 遵循上面的变化趋势下, 等级为 0 的变化最为剧烈。

另外, 如果运行队列的 load 值比当前 $\text{cpu_load}[]$ 的值大, 会对 M 的值有个补偿: 举这样一个例子, 假如 $M - m_i = 17$, 对于计算 $i=4$, 来说, $17/16 = 1$, 余数为 1, 这样太浪费了, 我要把余数也计算进来, 所以我要在 $M - m_i$ 处理时, 加上 $(16-1)$, 这样, 就不会浪费余数了。一句话: Round Up。

具体在系统中这个函数是如何变化的呢? 可以在终端看下: `cat /proc/sched_debu`, 可以看到不同 cpu 的相应数值. 我的 pc 有两个 cpu, 它们相应的与此函数相关的数据如下:

cpu#0, 2705.784 MHz

```
.nr_running      : 2
.load            : 2048
.nr_load_updates : 6651134
.cpu_load[0]     : 1024
.cpu_load[1]     : 576
.cpu_load[2]     : 364
.cpu_load[3]     : 214
.cpu_load[4]     : 124
```

...

cpu#1, 2705.784 MHz

```
.nr_running      : 0
.load            : 0
.nr_load_updates : 6846119
.cpu_load[0]     : 0
```

```
.cpu_load[1]      : 0
.cpu_load[2]      : 15
.cpu_load[3]      : 61
.cpu_load[4]      : 110
```

现在我在此函数里加一些调试信息，看它在系统 boot 过程中的变化，借此总结出此函数的变化趋势。 我加了一些调试信息,新的函数如下：

```
static void update_cpu_load(struct rq *this_rq)
{
    unsigned long this_load = this_rq->load.weight;
    int i, scale;

    this_rq->nr_load_updates++;
    printk("Before calculate: %d\n", this_load); //在调用此函数时，打印当前运行队列的负载
    /* Update our load: */
    for (i = 0, scale = 1; i < CPU_LOAD_IDX_MAX; i++, scale += scale) {
        unsigned long old_load, new_load;

        /* scale is effectively 1 << i now, and >> i divides by scale */

        old_load = this_rq->cpu_load;
        new_load = this_load;
        /*
         * Round up the averaging division if load is increasing. This
         * prevents us from getting stuck on 9 if the load is 10, for
         * example.
         */
        if (new_load > old_load)
            new_load += scale-1;
        this_rq->cpu_load = (old_load*(scale-1) + new_load) >> i;
    }
}
```

```
        printk(KERN_INFO "old_load = %d,this_rq->cpu_load[%d] = %d, new_load
= %d\n",old_load, i, this_rq->cpu_load, new_load);
        //这儿我分别打印了更新前的 cpu_load,更新后的 cpu_load,补偿后的运行队列负载
    }

}
```

下面是调试的信息(限于篇幅，只分析几个调试片段):

debug information:

```
[ 0.007812] Before calculate: 0
[ 0.007812] old_load = 0,this_rq->cpu_load[0] = 0, new_load = 0
[ 0.007812] old_load = 0,this_rq->cpu_load[1] = 0, new_load = 0
[ 0.007812] old_load = 0,this_rq->cpu_load[2] = 0, new_load = 0
[ 0.007812] old_load = 0,this_rq->cpu_load[3] = 0, new_load = 0
[ 0.007812] old_load = 0,this_rq->cpu_load[4] = 0, new_load = 0
...
[ 0.015625] Before calculate: 7266
[ 0.015625] old_load = 0,this_rq->cpu_load[0] = 7266, new_load =
7266
[ 0.015625] old_load = 0,this_rq->cpu_load[1] = 3633, new_load =
7267
[ 0.015625] old_load = 0,this_rq->cpu_load[2] = 1817, new_load =
7269
[ 0.015625] old_load = 0,this_rq->cpu_load[3] = 909, new_load =
7273
[ 0.015625] old_load = 0,this_rq->cpu_load[4] = 455, new_load =
7281
...
[ 0.015625] <MAX9724> GPIOs setup done
[ 0.023437] Before calculate: 177522
```



```
[ 0.023437] old_load = 7266,this_rq->cpu_load[0] = 177522, new_load
= 177522
[ 0.023437] old_load = 3633,this_rq->cpu_load[1] = 90578, new_load
= 177523
[ 0.023437] old_load = 1817,this_rq->cpu_load[2] = 45744, new_load
= 177525
[ 0.023437] old_load = 909,this_rq->cpu_load[3] = 22986, new_load =
177529
[ 0.023437] old_load = 455,this_rq->cpu_load[4] = 11522, new_load =
177537
...
[ 0.046875] Before calculate: 355044
[ 0.046875] old_load = 0,this_rq->cpu_load[0] = 355044, new_load =
355044
[ 0.046875] old_load = 22644,this_rq->cpu_load[1] = 188844,
new_load = 355045
[ 0.046875] old_load = 25731,this_rq->cpu_load[2] = 108060,
new_load = 355047
[ 0.046875] old_load = 17598,this_rq->cpu_load[3] = 59779, new_load
= 355051
[ 0.046875] old_load = 10125,this_rq->cpu_load[4] = 31683, new_load
= 355059
...
[ 0.078125] Before calculate: 0
[ 0.078125] old_load = 0,this_rq->cpu_load[0] = 0, new_load = 0
[ 0.078125] old_load = 23605,this_rq->cpu_load[1] = 11802, new_load
= 0
[ 0.078125] old_load = 45587,this_rq->cpu_load[2] = 34190, new_load
= 0
[ 0.078125] old_load = 40046,this_rq->cpu_load[3] = 35040, new_load
= 0
```

```
[ 0.078125] old_load = 26104,this_rq->cpu_load[4] = 24472, new_load
= 0
[ 0.078125] I think it only execute several times.....
...
[ 0.101562] Before calculate: 0
[ 0.101562] old_load = 0,this_rq->cpu_load[0] = 0, new_load = 0
[ 0.101562] old_load = 2950,this_rq->cpu_load[1] = 1475, new_load =
0
[ 0.101562] old_load = 19231,this_rq->cpu_load[2] = 14423, new_load
= 0
[ 0.101562] old_load = 26827,this_rq->cpu_load[3] = 23473, new_load
= 0
[ 0.101562] old_load = 21508,this_rq->cpu_load[4] = 20163, new_load
= 0

[ 0.132812] Before calculate: 0
[ 0.132812] old_load = 0,this_rq->cpu_load[0] = 0, new_load = 0
[ 0.132812] old_load = 184,this_rq->cpu_load[1] = 92, new_load = 0
[ 0.132812] old_load = 6084,this_rq->cpu_load[2] = 4563, new_load =
0
[ 0.132812] old_load = 15723,this_rq->cpu_load[3] = 13757, new_load
= 0
[ 0.132812] old_load = 16612,this_rq->cpu_load[4] = 15573, new_load
= 0
[ 0.140625] Before calculate: 0
[ 0.140625] old_load = 0,this_rq->cpu_load[0] = 0, new_load = 0
[ 0.140625] old_load = 92,this_rq->cpu_load[1] = 46, new_load = 0
[ 0.140625] old_load = 4563,this_rq->cpu_load[2] = 3422, new_load =
0
[ 0.140625] old_load = 13757,this_rq->cpu_load[3] = 12037, new_load
= 0
```

```
[ 0.140625] old_load = 15573,this_rq->cpu_load[4] = 14599, new_load
= 0
[ 0.156250] TCP reno registered
.....
[ 79.617187] Before calculate: 360213
[ 79.617187] old_load = 360213,this_rq->cpu_load[0] = 360213, new_load = 360213
[ 79.617187] old_load = 360213,this_rq->cpu_load[1] = 360213, new_load = 360213
[ 79.617187] old_load = 360213,this_rq->cpu_load[2] = 360213, new_load = 360213
[ 79.617187] old_load = 360213,this_rq->cpu_load[3] = 360213, new_load = 360213
[ 79.617187] old_load = 360213,this_rq->cpu_load[4] = 360213, new_load = 360213
[ 79.656250] Before calculate: 360213
.....
```

为了大家看得方便，我把无关 的启动信息用...代替了。

大家只需要看我添加的打印信息即可。

可以明显看到,运行队列的负载在变化 。

this_rq->load.weight 的变化规律：

初始化 0 -->突然变得很大 -->0---->最终的平衡值(如果没有新的 task 加入)

大家可以分析四种特殊情况：

- 1) 刚启动，运行队列负载为 0 -->都是 0
- 2) 一些进程加入，运行队列负载突增 -->按照公式，分导递增
- 3) 运行队列负载又变成 0 -->按照公式，分导递减
- 4) 运行队列负载逐渐趋于平衡 → 所有层的负载与运行队列负载相同

通过后面的调试发现，cpu_load[]数组的值，总是在尽量地接近运行队列的 load 值，，随着 scale 增大，接进得越慢。。。cpu_load[0]总是与运行队列的 load 值相同。

后面的文章里，我将会总体上分析不同 cpu 之间调度平衡的原理及代码执行实例。

参考资料：

[\[PATCH -cfs\] Fix cpu_load calculation error](#)

(<http://lists.zerezo.com/linux-kernel/msg12735442.html>)

[CFS-devel, group scheduler, fixes](#)

(<http://lists.zerezo.com/linux-kernel/msg13412736.html>)

[Re: \[PATCH\] sched: Improve readability in update_cpu_load\(\) code](#)

(<http://lkml.indiana.edu/hypermail/linux/kernel/0805.1/3068.html>)

Linux 下 SVN+apache 简单安装与配置

ChinaUnix 网友: liaosnet

前言:

最近整理以前写的一些 SHELL 脚本,总是发现改得一塌糊涂.于是乎决定找个工具来管理一下,在需要更改的时候更改,改完保存起来.以防止使用了错误的版本,保证版本最新。

在网上找到了 SVN 这样东东,觉得功能还好.同时搜索了安装方式,竟然都是 configure && make && make install 的方式.我这样的懒人可不喜欢为了那些依赖关系而烦恼,于是找到了一下编译好了的版本.^_^

相关平台软件:

这里使用的操作系统是 suse10(虽然 CollabNet Subversion 是在 rhel4/5 下编译的,但仍可在 suse10 上使用).

CollabNet Subversion v1.5.6

CollabNetSubversion-client-1.5.6-1.i386.rpm --客户端

CollabNetSubversion-server-1.5.6-1.i386.rpm --服务端

CollabNetSubversion-extras-1.5.6-1.i386.rpm --扩展件

下载地址:

<http://www.open.collab.net/downloads/subversion/>

TortoiseSVN 1.6.1

TortoiseSVN-1.6.1.16129-win32-svn-1.6.1.msi --TortoiseSVN,这里当做 windows 客户端使用

下载地址:

<http://tortoisesvn.net/downloads>

安装过程:

安装已经编译的 rpm 即可,很容易.(这也是我偷懒的原因,省事! ^_^).

先安装 client,然后 server,最后是 extras.

```
# rpm -ivh CollabNetSubversion-client-1.5.6-1.i386.rpm
```

```
warning: CollabNetSubversion-client-1.5.6-1.i386.rpm: V3 DSA signature: NOKEY, key ID 35bcca43
```

```
Preparing... #####
```



```
[100%]
  1:CollabNetSubversion-
cli##### [100%]

# rpm -ivh CollabNetSubversion-server-1.5.6-1.i386.rpm
warning: CollabNetSubversion-server-1.5.6-1.i386.rpm: V3 DSA signature: NOKEY, key ID
35bcca43
Preparing... #####
[100%]
  1:CollabNetSubversion-
ser##### [100%]

# rpm -ivh CollabNetSubversion-extras-1.5.6-1.i386.rpm
warning: CollabNetSubversion-extras-1.5.6-1.i386.rpm: V3 DSA signature: NOKEY, key ID
35bcca43
Preparing... #####
[100%]
  1:CollabNetSubversion-
ext##### [100%]
```

三个 rpm 安装安装完成了..系统增加了一个 csvn 用户/组.

```
# more /etc/passwd
csvn:x:1000:1000:CollabNet Subversion:/opt/CollabNet_Subversion:/bin/bash
# more /etc/group
csvn:!:1000:
```

创建了目录/opt/CollabNet_Subversion 用于存放 SVN 相关的程序.

```
# ll /opt/CollabNet_Subversion
drwxr-xr-x  4 root root  960 Apr 20 14:25 bin
drwxr-xr-x  2 root root  312 Apr 20 14:24 build
drwxr-xr-x  2 root root  216 Apr 20 14:24 docs
drwxr-xr-x  6 root root 3424 Apr 20 14:25 lib
drwxr-xr-x  2 root root  232 Apr 20 14:24 licenses
drwxr-xr-x  2 root root  184 Apr 20 14:25 modules
drwxr-xr-x  2 root root  208 Apr 20 14:25 sbin
drwxr-xr-x  4 root root   96 Feb 27 01:17 share
```

配置 SVN:

编译好的 SVN 配置起来也很简单(感谢编译人员辛勤的劳动).在/opt/CollabNet_Subversion/bin 目录下执行 Configure-CollabNet-Subversion ,按照提示做就行了.(表示操作)

```
# ./Configure-CollabNet-Subversion
```

CollabNet Subversion Server is a free download of open-source Subversion, compiled and tested by CollabNet. For more information about CollabNet Subversion, visit the CollabNet community at <http://open.collab.net>.

Would you like to configure the CollabNet Subversion Server now? [yes] (回车)

Welcome to CollabNet Subversion Server setup.

You will now be asked some questions to configure CollabNet Subversion. Defaults are given in square brackets. If a default is okay for you Just hit [Enter] to continue.

This setup can configure either Apache or svnserve as Subversion server, for you. Would you like to configure Apache as your Subversion server now? [yes] (回车)

Continuing with apache Subversion server configuration ...

Specify the base directory for Subversion repositories: [/var/svn/repositories]
/opt/svn/repositories (这里我更改了路径,不使用默认的.)

Non-existing base path, Let me create and continue [yes] (回车)

Base path created.

Would you like to create a new Subversion repository? [yes] (回车)

What should be your new Subversion repository name? test (这里建立一个名为 test 的库)

New repository created.

Would you like to initialize the repository with trunk/branches/tags folders? [no] (回车)

Done.

Would you like to create another Subversion repository? [yes] (回车)

What should be your new Subversion repository name? test2 (这里再建立一个名为 test 的库)

New repository created.

Would you like to initialize the repository with trunk/branches/tags folders? [no](回车)(暂时不导入初始数据)

Done.

Would you like to create another Subversion repository? [yes] no (已经建立了 2 个库,不建了.)

Would you like to give the name and port that the server uses to identify itself? [yes]

Registered DNS name or IP address: 192.168.80.100 (这里是 apache 的配置,apache 的服务 IP)

Server port number: 8000 (apache 的服务端口号)

Allow anonymous read access? [no] (回车)

0 users currently have access to this repository:

Would you like to create a new Subversion account? [yes] (回车)

Account username

(To re-create or delete a Subversion account, enter an existing username): svn (建立一个 SVN 帐号)

Account password: (密码)

Retype account password: (重复密码)

Adding password for user svn

Account created

Would you like to create another user account? [yes] no (不建立另一用户)

Would you like to configure ViewVC? [yes] (启用 ViewVC,回车)

Stopping CollabNet Subversion: httpd (no pid file) not running
done

Starting CollabNet Subversion: done

(完成,最后的提示信息)

Congratulations! You have successfully configured the CollabNet Subversion server.

You can access your repositories at the following URL:

http://192.168.80.100:8000/svn/YOUR-REPOS-NAME

http://192.168.80.100:8000/viewvc/YOUR-REPOS-NAME

配置完成后检查:

与SVN相关的APACHE已经起来了.

```
# ps -ef | grep httpd
```

```
root    6989    1  2 14:28 ?        00:00:00 /opt/CollabNet_Subversion/bin/httpd -D
csvn_installed -D viewvc_installed -k start
csvn    6990    6989  0 14:28 ?        00:00:00 /opt/CollabNet_Subversion/bin/httpd -D
csvn_installed -D viewvc_installed -k start
csvn    6991    6989  0 14:28 ?        00:00:00 /opt/CollabNet_Subversion/bin/httpd -D
csvn_installed -D viewvc_installed -k start
csvn    6992    6989  0 14:28 ?        00:00:00 /opt/CollabNet_Subversion/bin/httpd -D
csvn_installed -D viewvc_installed -k start
csvn    6993    6989  0 14:28 ?        00:00:00 /opt/CollabNet_Subversion/bin/httpd -D
csvn_installed -D viewvc_installed -k start
csvn    6994    6989  0 14:28 ?        00:00:00 /opt/CollabNet_Subversion/bin/httpd -D
csvn_installed -D viewvc_installed -k start
csvn    6995    6989  0 14:28 ?        00:00:00 /opt/CollabNet_Subversion/bin/httpd -D
csvn_installed -D viewvc_installed -k start
```

在浏览器中,YOUR-REPOS-NAME 换成建立的库名,这里为 test,test2..

<http://192.168.80.100:8000/svn/YOUR-REPOS-NAME>

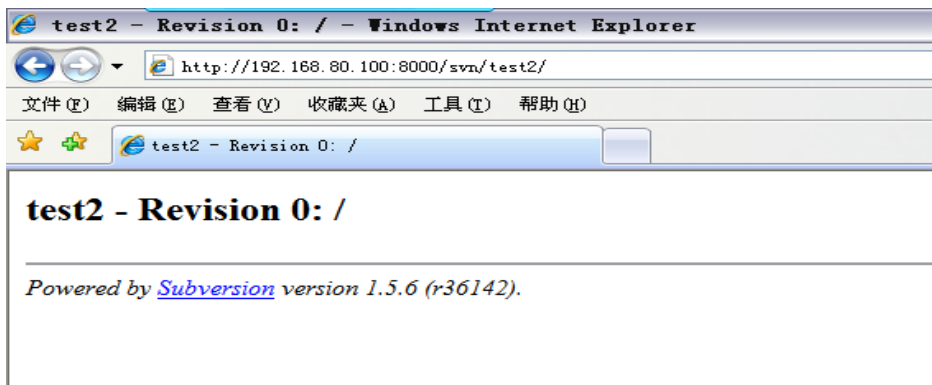
<http://192.168.80.100:8000/viewvc/YOUR-REPOS-NAME>

注意浏览器上的地址栏。

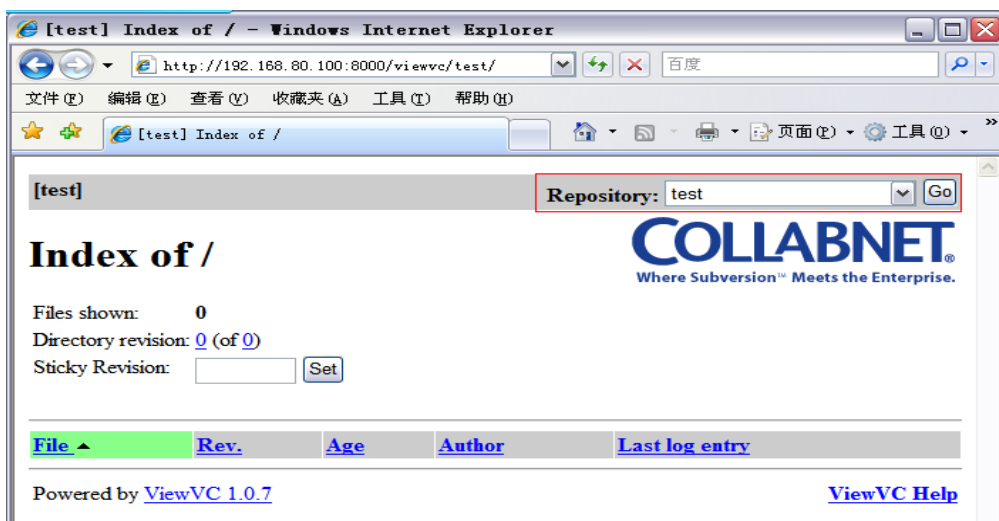
Test:



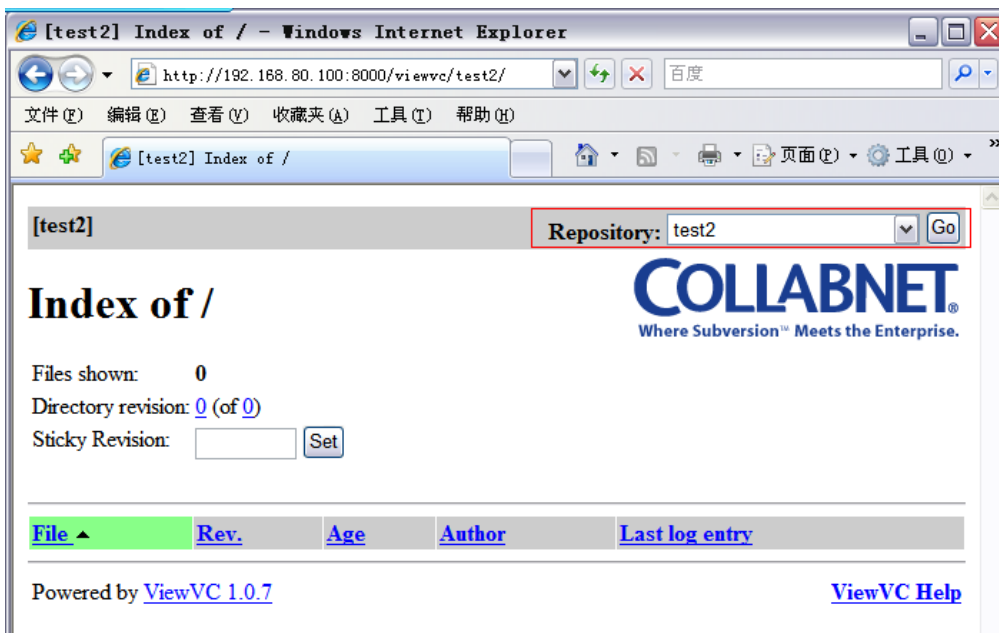
test2:



ViewVC test:



ViewVC test2:



个人感觉还是 ViewVC 实用些...

这时,由于还没有导入初始数据..所以这里的显示均为 0.

库导入:

在/root/tmp/下建了一个测试脚本 test.sh,内容很简单.

```
#!/bin/sh
```

```
echo "test"
```

现在把/root/tmp 导入到 SVN 库 test 里

```
# /opt/CollabNet_Subversion/bin/svn import /root/tmp file:///opt/svn/repositories/test -m "SVN test"
```

Adding /root/tmp/test.sh

Committed revision 1.

导入完成,增加一个文件.当前的版本为 1.

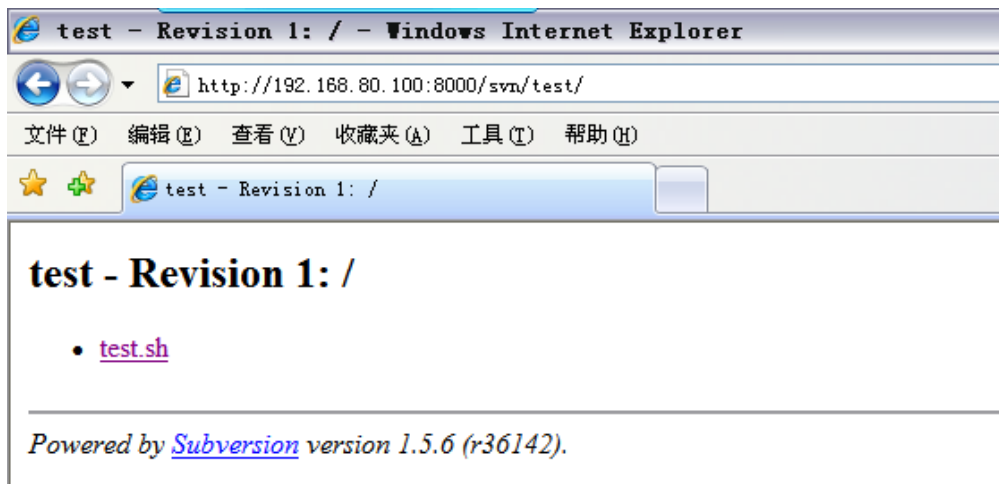
检查导入

```
# /opt/CollabNet_Subversion/bin/svn list file:///opt/svn/repositories/test
```

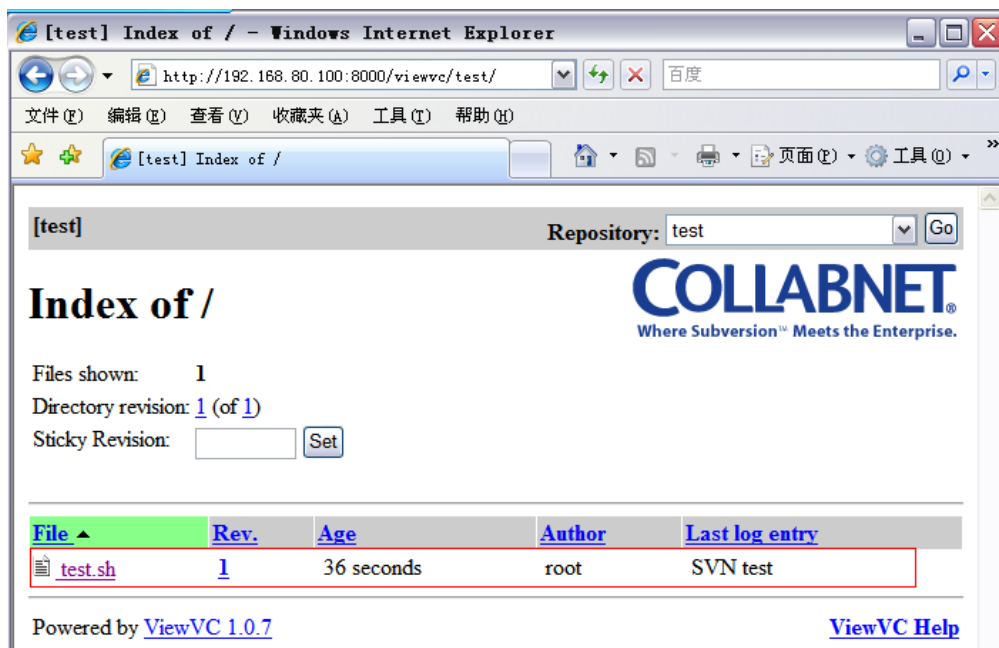
test.sh

在浏览器中查看。

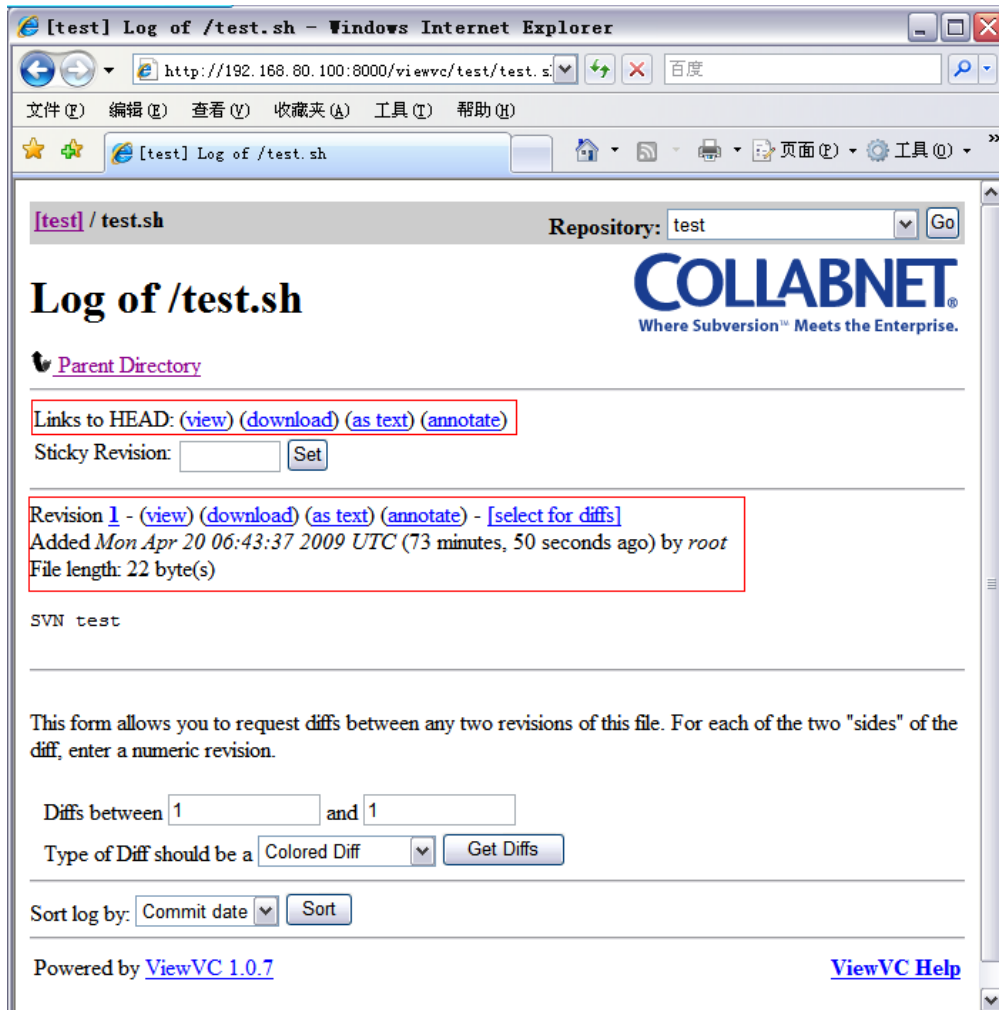
test:



ViewVc test:



点开 test.sh,可以看到一些信息。



OKey..SVN 库建立,数据初始化合算是完成了..下面来点简单点的更改..这里就用到了 TortoiseSVN 这个 windows 版的 svn 工具当客户端用(linux 也有客户端,也就是刚我们安装的那个 client,可是那个 command line....呵呵).

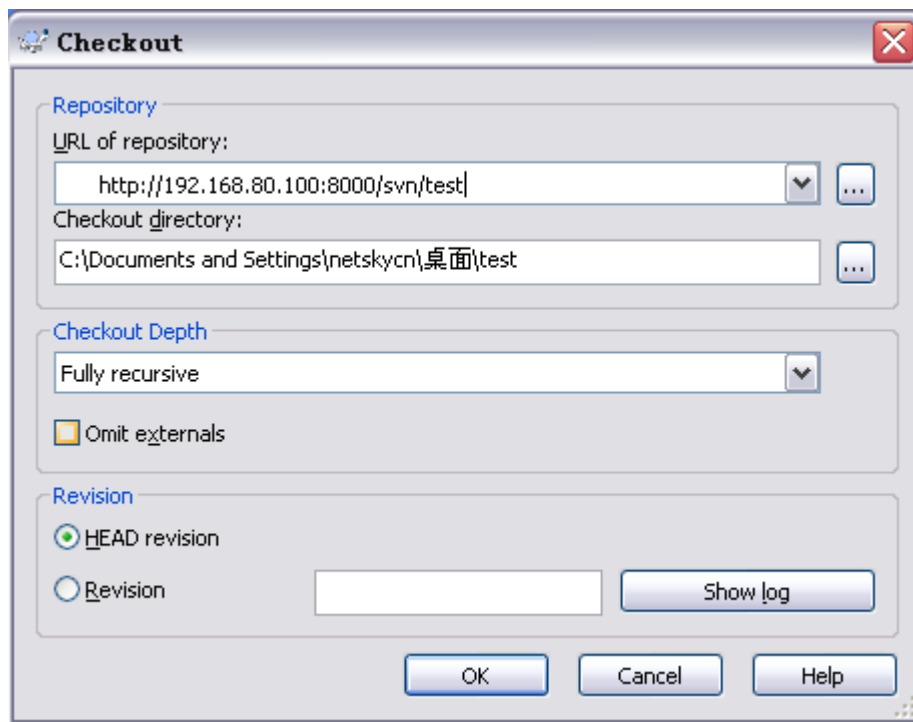
TortoiseSVN 的安装就不说了,windows 下的东东,基本上一路 next..完成后,重启一下系统,然后就 OK 了.

现在右键看一下..^_^ ,这就是增加的 TortoiseSVN 的功能..我们这用到的要少些(当客户端用嘛).

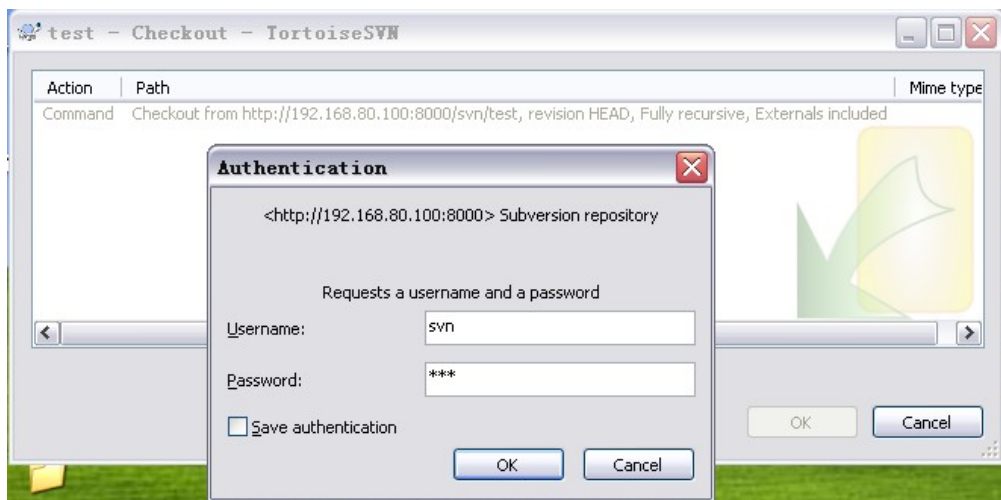


1.检出(SVN checkout).

点 SVN Checkout,输出要检出的库/目录/文件



使用 SVN 帐号(上面建立的 svn 帐号)



完成.OK

在 Checkout 目录生成一个指定目录...包含里初始的 test.sh.



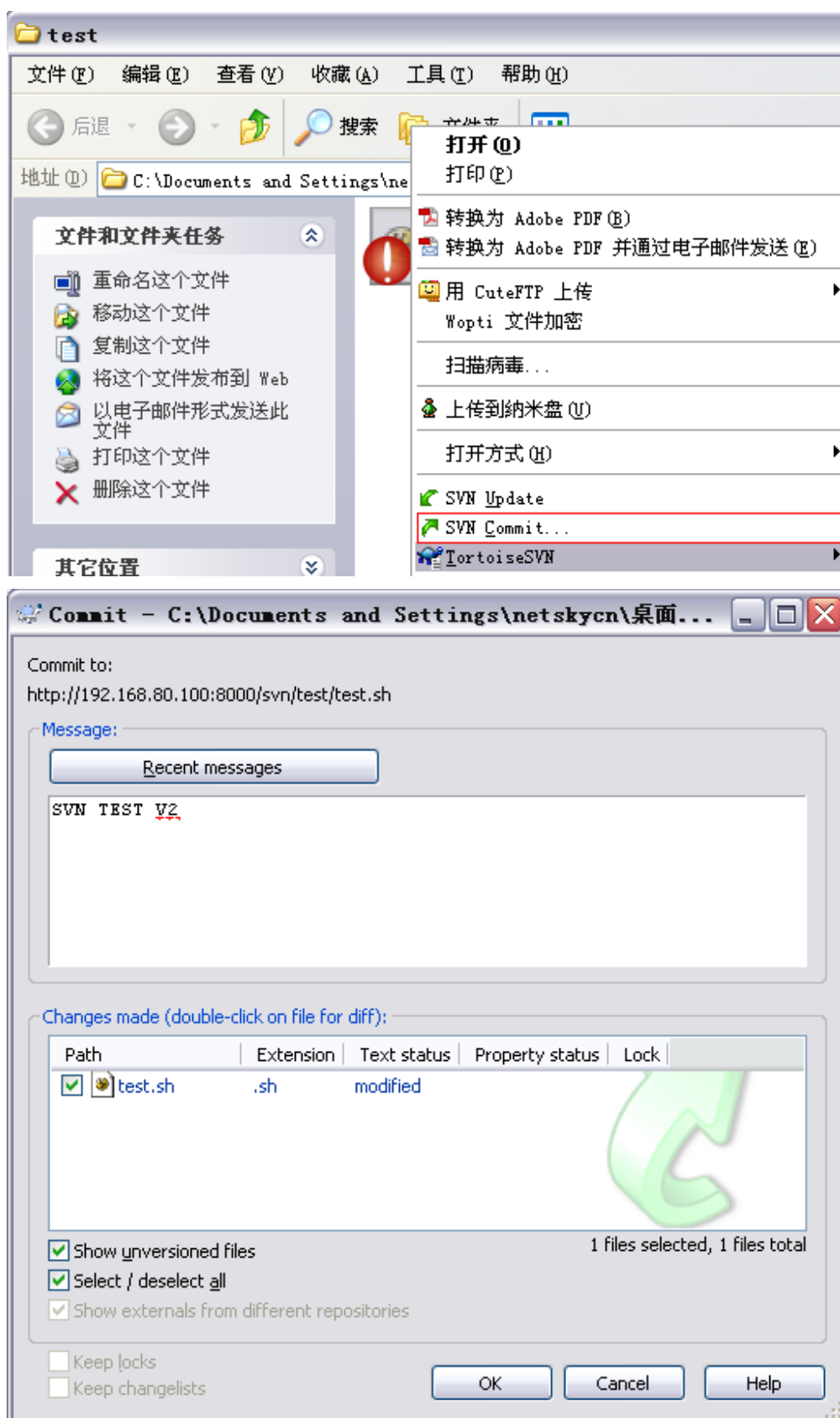
2.更改

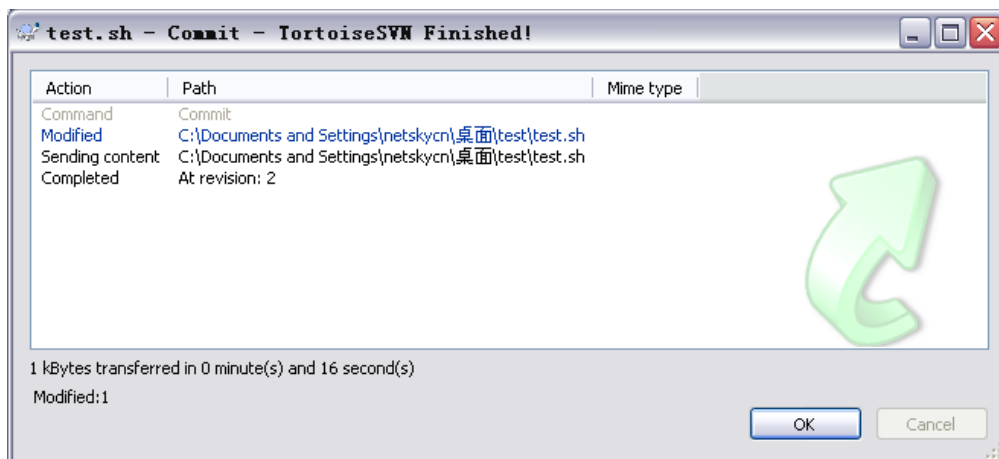
修改 test.sh 文件内容后,保存..图标变了,表示已经更改了..



3.确认提交(SVN Commit)

右键 SVN Commit

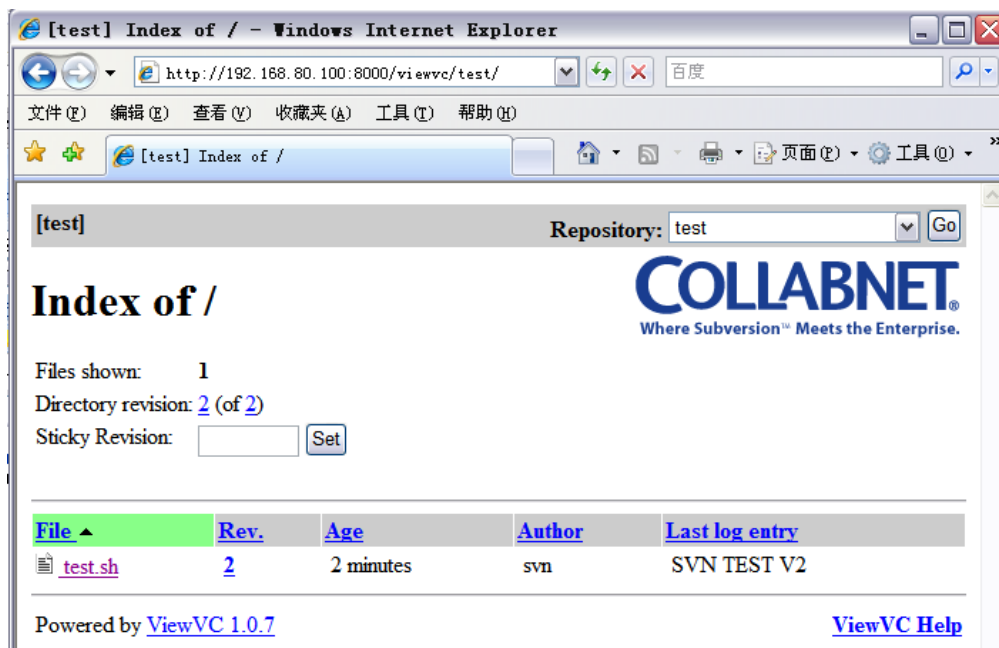




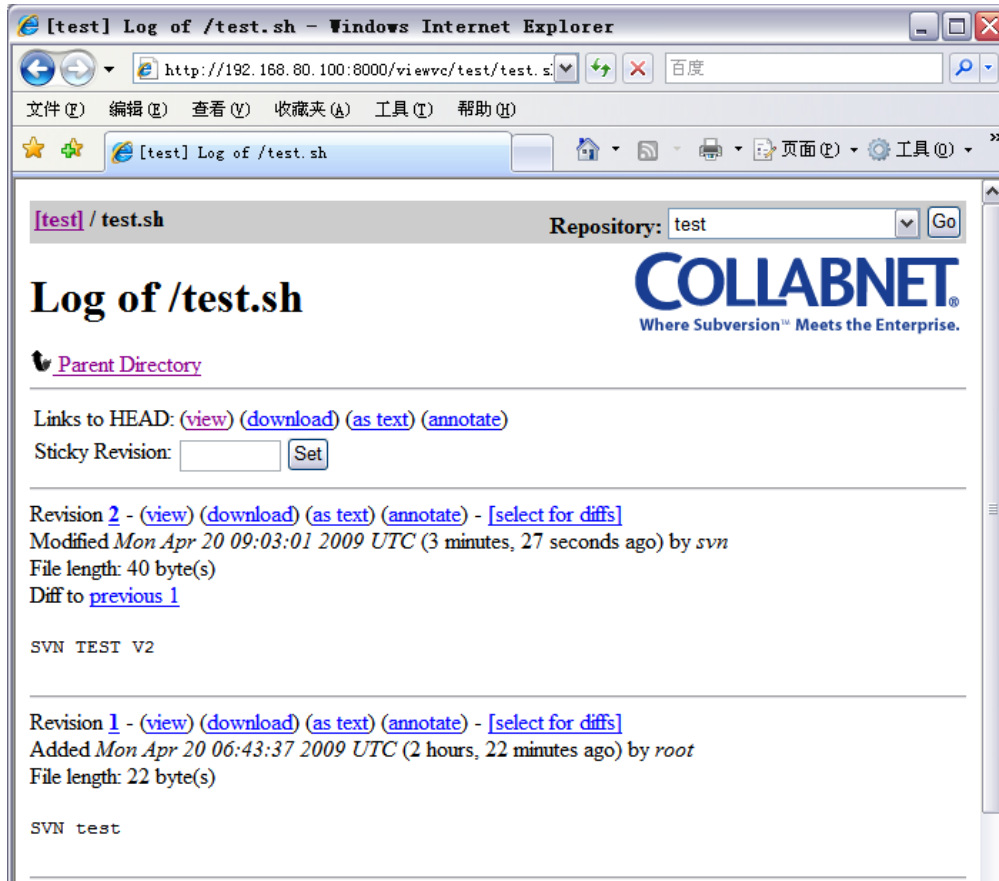
完成后,本地恢复..

4.检查更新.

在浏览器上查看更新..已经变成版本 2.



test.sh 更改详细信息。



小结:

SVN 是个很好的版本控制软件。使用它后，可以容易地保证是在最新的版本上进行的更改。

DirectFB 之一：PC 平台编译与安装

ChinaUnix 网友：oldstorm

上班的时候，经常听旁边的一个专门做图形库的小组提起 DirectFB, 据说今后的几年，DirectFB 将取代现有图形库，并且采用 DirectFB + GTK 形式，支持现有的很多基于 GTK 开源软件。在国内的一些论坛上，比如 MiniGUI.org 上，也看到过他们在做 MiniGUI + DirectFB 的文章，虽然现在不知道做到什么程度了，但是可以看出，DirectFB 还是有它的优势的。

最近周末正好有时间，所以也想玩玩。大概的计划是：PC 机上测试 -> ARM 开发板上测试 -> ARM 开发板 gfxdriver 开发。

1. DirectFB 相关网站

[DirectFB 官方主页](#) [资料下载/文档](#)

[DirectFB Wiki Page](#) 很多 DirectFB 的文档都可以在这里找到

2. 开发环境

Linux: CentOS 5.1 (Kernel 2.6.18-53.el5xen)

GCC: 4.1.2

3. 下载 DirectFB 源代码

DirectFB 提供 3 种形式的下载方式: git, cvs 和压缩包。我一般使用 git, 代码比较方便。

```
# git clone git://git.directfb.org/git/directfb/core/DirectFB.git
```

另外, 出了 DirectFB 库之外, 如果要支持 multi application core, 还需要 fusion 内核驱动模块; 为了测试 DirectFB, 可以下载现有的测试程序:

[Fusion: linux-fusion-8.0.3.tar.gz](http://linux-fusion-8.0.3.tar.gz)

Example: DirectFB-examples-1.2.0.tar.gz

4. 编译和安装 fusion

```
# cd ~/DirectFB
# tar xzf linux-fusion-8.0.3.tar.gz
# cd linux-fusion-8.0.3
# make
# sudo make install
```

创建 fusion 设备文件:

```
# sudo mkdir /dev/fusion
# sudo mknod /dev/fusion/0 c 250 0
# sudo mknod /dev/fusion/1 c 250 1
# sudo mknod /dev/fusion/2 c 250 2
```

注意: 要编译 fusion 模块, 需要安装内核源码。fusion 模块默认安装到 /lib/modules/`uname -r`/ 下。

最新的源程序可以用 git 下载:

```
# git clone git://git.directfb.org/git/directfb/core/linux-fusion.git
```

使用这个版本的 fusion 的话, 编译 DirectFB 的时候不会提示 ../../lib/fusion/types.h:42:2: error: #error Major version of Fusion Kernel Module too low! Upgrade your kernel. 的错误。

5. 编译 DirectFB 库

```
# cd ~/DirectFB/DirectFB
# ./autogen.sh \
  --prefix=/usr/local \
  --disable-osx \
  --enable-x11 \
```



```
--disable-fbdev      \
--disable-sdl        \
--disable-vnc        \
--with-gfxdrivers=none \
--with-inputdrivers=linuxinput \
--enable-multi       \
--enable-debug
```

```
# make
# sudo make install
```

* 我一般将这些命令写到一个 shell 脚本里面。如(build-for-pc.sh)

6. 编译错误处理

编译 DirectFB 的时候，提示一下错误：

```
# make
.././../lib/fusion/types.h:42:2: error: #error Major version of Fusion Kernel Module too low!
Upgrade your kernel.
```

这是因为：

```
--- /usr/include/linux/fusion.h 文件---
#define FUSION_API_MAJOR    8      /* Increased if backward compatibility is dropped.
*/
#define FUSION_API_MINOR    0      /* Increased if new features are added. */
#define FUSION_API_MICRO    3      /* Micro release. Bugfixes. */

--- DirectFB/lib/fusion/types.h ---

#define FUSION_API_MAJOR_REQUIRED 8
#define FUSION_API_MINOR_REQUIRED 0

#if FUSION_API_MAJOR_REQUIRED > FUSION_API_MAJOR_PROVIDED
#error Major version of Fusion Kernel Module too low! Upgrade your kernel.
#else
...
#endif
```

可以看出，FUSION_API_MAJOR_PROVIDED 没有在 /usr/include/linux/fusion.h 定义，所以只需要将 DirectFB/lib/fusion/types.h 中 FUSION_API_MAJOR_PROVIDED 的改成 FUSION_API_MAJOR 即可。修改完之后是：

```
# git diff lib/fusion/types.h
```

```
-----
diff --git a/lib/fusion/types.h b/lib/fusion/types.h
index 1b862a4..3776e93 100644
--- a/lib/fusion/types.h
+++ b/lib/fusion/types.h
@@ -38,11 +38,11 @@
#define FUSION_API_MAJOR_REQUIRED 8
#define FUSION_API_MINOR_REQUIRED 0

-#if FUSION_API_MAJOR_REQUIRED > FUSION_API_MAJOR_PROVIDED
+#if FUSION_API_MAJOR_REQUIRED > FUSION_API_MAJOR
#error Major version of Fusion Kernel Module too low! Upgrade your kernel.
#else
-#if FUSION_API_MAJOR_REQUIRED == FUSION_API_MAJOR_PROVIDED
-#if FUSION_API_MINOR_REQUIRED > FUSION_API_MINOR_PROVIDED
+#if FUSION_API_MAJOR_REQUIRED == FUSION_API_MAJOR
+#if FUSION_API_MINOR_REQUIRED > FUSION_API_MINOR
#error Minor version of Fusion Kernel Module too low! Upgrade your kernel.
#endif
#endif
-----
```

再编译，OK。DirectFB 将被安装到 prefix 指定的地方(/usr/local/lib/)。

7. 添加 /usr/local/lib/ 路径

为了使用 DirectFB，需要将 DirectFB 库所在的路径添加到 系统搜索路径。否则：

- 1) pkg-config 找不到 DirectFB
- 2) ld 命令也找不到 DirectFB 库

所以：

```
# sudo vi /etc/ld.so.conf.d/qt-i386.conf
```

添加一行： /usr/local/lib

```
# sudo ldconfig
```

8. 编译和安装示例程序

```
# tar xzf DirectFB-examples-1.2.0.tar.gz
# cd DirectFB-examples-1.2.0
# ./configure \
  --prefix=/usr/local \
  --enable-debug
```

```
# make
# sudo make install
```

示例程序如 df_andi 将被安装到 /usr/local/bin

9. 运行示例程序

```
# sudo depmod -a
# sudo modprobe fusion # 安装 fusion 内核模块
# cd /usr/local/bin
# sudo ./df_andi
```

```
~~~~~ | DirectFB 1.3.0 | ~~~~~
(c) 2001-2008 The world wide DirectFB Open Source Community
(c) 2000-2004 Convergence (integrated media) GmbH
-----
```

```
(*) DirectFB/Core: Multi Application Core. (2009-03-15 08:52) [ DEBUG ]
(*) Fusion/SHM: NOT using MADV_REMOVE (2.6.18.0 < 2.6.19.2)! [0x02061200]
(*) Direct/Thread: Started 'Fusion Dispatch' (-1) [MESSAGING OTHER/OTHER 0/0]
<10485760>...
(*) Direct/Thread: Started 'Linux Input' (-1) [INPUT OTHER/OTHER 0/0] <10485760>...
(*) DirectFB/Input: AT Translated Set 2 keyboard (1) 0.1 (directfb.org)
(*) Direct/Thread: Started 'Linux Input' (-1) [INPUT OTHER/OTHER 0/0] <10485760>...
(*) DirectFB/Input: HID 413c:3010 (2) 0.1 (directfb.org)
(*) Direct/Thread: Started 'Linux Input' (-1) [INPUT OTHER/OTHER 0/0] <10485760>...
(*) DirectFB/Input: PC Speaker (3) 0.1 (directfb.org)
(*) Direct/Thread: Started 'X11 Input' (-1) [INPUT OTHER/OTHER 0/0] <10485760>...
(*) DirectFB/Input: X11 Input 0.1 (directfb.org)
(*) DirectFB/Genefx: MMX detected and enabled
(*) DirectFB/Graphics: MMX Software Rasterizer 0.6 (directfb.org)
(*) DirectFB/Core/WM: Default 0.3 (directfb.org)
(*) X11/Window: Creating 1280x1024 RGB32 window...
(*) X11/Display: Using XShm.
(*) X11/Window: Creating 1280x1024 RGB32 window...
(*) X11/Display: Using XShm.
(*) Direct/Interface: Loaded 'FT2' implementation of 'IDirectFBFont'.
(*) Direct/Interface: Loaded 'PNG' implementation of 'IDirectFBImageProvider'.
(*) Direct/Interface: Loaded 'JPEG' implementation of 'IDirectFBImageProvider'.
```

hoho, 无数的企鹅出现了~~~~~



10. 总结

- 1) DirectFB 支持非 Framebuffer 模式, 不使用 Framebuffer 也可以使用 DirectFB.
- 2) 一定要记得添加 /usr/local/bin 到 /etc/ld.so.conf.d/ 里。一开始忘了, 出现了一下错误:

错误 A: 编译示例程序的时候

.....

```
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking for DIRECTFB... no
configure: error: Requested 'directfb >= 1.2.0' but version of DirectFB is 1.0.1
*** DirectFB 1.2.0 or newer is required. The latest version
*** of DirectFB is available from http://www.directfb.org/.
make: *** No targets specified and no makefile found. Stop.
```

关于 pkg-config 命令是如何工作的, 参考 [pkg-config 工具](#).

错误 B: 运行的时候

```
# sudo ./df_andi
```

```
./df_andi: error while loading shared libraries: libdirectfb-1.3.so.0: cannot open shared
object file: No such file or directory
```

```
# ldd df_andi
```

```
linux-gate.so.1 => (0x0075e000)
libdirectfb-1.3.so.0 => not found
libfusion-1.3.so.0 => not found
libdirect-1.3.so.0 => not found
libpthread.so.0 => /lib/i686/nosegnet/libpthread.so.0 (0x0036d000)
libc.so.6 => /lib/i686/nosegnet/libc.so.6 (0x001f9000)
```

/lib/ld-linux.so.2 (0x001d7000)

```
#!/bin/sh
#
# Build DirectFB library for PC
#
# ChangeLog:
# V0.1 2009/03/15 Zeng Xianwei (xianweizeng@gmail.com)
#

prefix=/usr/local
def_options="--prefix=${prefix}"

fatal() {
    echo "Error: $@"
    exit 1
}

build_and_install() {
    ./autogen.sh \
    ${def_options} \
    --disable-osx \
    --enable-x11 \
    --disable-fbdev \
    --disable-sdl \
    --disable-vnc \
    --with-gfxdrivers=none \
    --with-inputdrivers=linuxinput \
    --enable-multi \
    --enable-debug

    make || fatal "make failed"

    sudo make install || fatal "make install failed"
}

build_and_install

exit 0
```


OpenBSD. Nginx. MySQL. PHP 环境搭建手册 (v2)

ChinaUnix 网友: young_king

前言

感谢 ChinaUnix 以及 ChinaUnix 论坛中的众多高手，在这里学到了很多！

本手册是 O.N.M.P.系列的第二版，在第一版手册中，只是简单的说明了 O.N.M.P.环境的基本搭建方法，并没有过多的考虑到生产系统适用的情况。第二版将作较大的修正，以使得本手册可以对生产环境的搭建起到有益的参考作用。

本人水平有限，手册中出现纰漏在所难免，您若在生产实际中应用时发现有什么问题，或者有什么更好的建议，请移步至 ChinaUnix 论坛的 BSD 版，或至偶的个人博客

(<http://onmp.cublog.cn>)，跟帖进行指正。希望大家的共同努力下，能够把本手册尽量做到完美！

本手册以在 OpenBSD 4.4 环境下搭建 Nginx、MySQL、PHP 环境为例进行讲解。按照惯例，root 权限。

一、OpenBSD 的安装及注意事项

OpenBSD 的安装这里就不多说了，主要是安装时系统组件的选择、分区和系统服务的部分。对于系统组件部分，作为生产系统来说，偶并不推荐安装 comp44.tgz 这个组件。因为 comp44.tgz 实际就是编译器，不装这个，可以在很大程度上避免安装一些非授权的软件，从而提高远程主机的安全性。如果真的需要安装什么软件，也可以在非重要的机器上进行编译，然后使用 PSFTP 等软件上传到远程主机上使用。关于 PSFTP 软件的使用，二楼有详细的介绍。

系统分区时，推荐将 /usr、/usr/local、/var、/var/mysql、/var/mail、/var/log、/var/nginx 等分区单独分出来，也就是说，进行比较细致的分区，防止某个目录中的文件膨胀占满整个分区导致的死锁等问题。假定你有一个 80G 的硬盘，一个分区示例在下面：

/	200M
(swap)	1G
/tmp	200M
/usr	500M
/usr/local	200M
/var	100M
/var/mysql	10G
/var/mail	100M

/var/log	5G~10G
/home	200M
/var/nginx	剩余空间

至于系统服务部分，建议所有的服务都选择“n”，亦即不随系统启动。

二、系统性能调优

对于多核的机器，使用 bsd.mp 这个核心。

```
mv /bsd /obsd
mv /bsd.mp /bsd
```

I、/etc/fstab 调优

```
vi /etc/fstab
```

在文件系统描述符部分，加入"noatime"和"softdep"。示例如下：

```
/dev/wd0a / ffs rw,noatime,softdep 1 1
/dev/wd0l /home ffs rw,nodev,nosuid,noatime,softdep 1 2
/dev/wd0d /tmp ffs rw,nodev,nosuid,noatime,softdep 1 2
/dev/wd0e /usr ffs rw,nodev,noatime,softdep 1 2
/dev/wd0f /usr/local ffs rw,nodev,noatime,softdep 1 2
/dev/wd0g /var ffs rw,nodev,nosuid,noatime,softdep 1 2
/dev/wd0i /var/log ffs rw,nodev,nosuid,noatime,softdep 1 2
/dev/wd0j /var/mail ffs rw,nodev,nosuid,noatime,softdep 1 2
/dev/wd0h /var/mysql ffs rw,nodev,nosuid,noatime,softdep 1 2
/dev/wd0k /var/nginx ffs rw,nodev,nosuid,noatime,softdep 1 2
```

友情提醒：softdep 是一种非同步的文件系统，意外掉电可能造成数据的遗失/损坏，生产系统请谨慎使用！

改完后保存退出，reboot，看能不能正常启动。

一般情况下是没有问题的，个别机器或虚拟机可能会因为兼容性的缘故，无法启动。那么就把根目录的 softdep 拿掉，其他目录保留，仍然会有作用。

更多关于 OpenBSD 环境下磁盘性能调优的内容，请参看偶博客的文章，地址在下面：

http://blog.chinaunix.net/u2/81136/showart_1841280.html

II、/etc/sysctl.conf 调优

```
vi /etc/sysctl.conf
```

跳到最后，加入下面的内容：

```
# 增大文件系统缓存到 1M
kern.maxvnodes=131072
```

```
# 允许最多 65536 个进程
kern.maxproc=65536
```

```
# 同时最多打开 65536 个文件
kern.maxfiles=65536
```

```
# 并发连接最大 65536
kern.somaxconn=65536
```

```
# 保留的最少连接数
kern.sominconn=256
kern.maxclusters=32768
```

```
# 增大 TCP 接收/发送缓存到 64K
net.inet.tcp.recvspace=65536
net.inet.tcp.sendspace=65536
```

```
# 增大 UDP 接收/发送缓存到 64K
net.inet.udp.recvspace=65536
net.inet.udp.sendspace=65536
```

注意：虚拟机测试只加最上面一行 kern.maxvnodes=65536 即可，其他的不必加了，否则会有各种问题。独立机器的可以加上。

保存退出，reboot。不能正常启动的就把除 kern.maxvnodes 以外的数字调小或者禁用再试。

三、OpenBSD.Nginx.MySQL.PHP 软件环境的安装

OpenBSD 环境下软件的安装是非常简单的，因为在官方的 ftp 中提供了已经编译好的二进制包，需要安装的软件都在 ftp 中，从 ftp 中安装即可。

小提示：如果你机器比较多，你可以把需要安装的软件包都 down 回来，其他机器再来这里安装，速度会非常快！

```
export PKG_PATH=ftp://ftp.openbsd.org/pub/OpenBSD/4.4/packages/i386/
```

```
pkg_add mysql-server php5-fastcgi php5-gd-5.2.6-no_x11 phpMyAdmin lighttpd-1.4.19p3
```

nginx pecl-APC

以 偶的 1M ADSL 小水管为例，大约也就二十分钟左右就安装完毕了！真的是非常快！和其他系统的 wget 源码、./configure && make && make install...所需要的时间相比，效率是非常高！而且，由于 OpenBSD 默认采用比较高的安全策略，装上的环境安全性也比其他系统要高！

等所需的软件都安装完成后，按提示作链接并创建 PHP 临时工作目录：

```
ln -s /var/www/conf/modules.sample/php5.conf /var/www/conf/modules
ln -fs /var/www/conf/php5.sample/apc.ini /var/www/conf/php5/apc.ini
ln -fs /var/www/conf/php5.sample/gd.ini /var/www/conf/php5/gd.ini
ln -fs /var/www/conf/php5.sample/mbstring.ini /var/www/conf/php5/mbstring.ini
ln -fs /var/www/conf/php5.sample/mcrypt.ini /var/www/conf/php5/mcrypt.ini
ln -fs /var/www/conf/php5.sample/mysql.ini /var/www/conf/php5/mysql.ini
mkdir /var/nginx/sesstmp
chmod 0777 /var/nginx/sesstmp
```

pkg_info 检查一下系统中安装了些什么软件包：

```
# pkg_info
gettext-0.17      GNU gettext
jpeg-6bp3         IJG's JPEG compression utilities
libconv-1.12      character set conversion library
libltdl-1.5.26    GNU libtool system independent dlopen wrapper
libmcrypt-2.5.7p1 interface to access block/stream encryption algorithms
libxml-2.6.32p1   XML parsing library
lighttpd-1.4.19p3 secure, fast, compliant, and very flexible web-server
lua-5.1.3         powerful, light-weight programming language
mhash-0.9.1p1     strong hash library
mysql-client-5.0.51ap0 multithreaded SQL database (client)
mysql-server-5.0.51ap1 multithreaded SQL database (server)
nginx-0.6.31      robust and small HTTP server and mail proxy server
p5-DBD-mysql-4.005 MySQL drivers for the Perl DBI
p5-DBI-1.604      unified perl interface for database access
p5-Net-Daemon-0.43 extension for portable daemons
p5-PlRPC-0.2018p0 module for writing rpc servers and clients
pcre-7.7p0       perl-compatible regular expression library
pecl-APC-3.0.19   Alternative PHP Cache
php5-core-5.2.6   server-side HTML-embedded scripting language
php5-fastcgi-5.2.6 stand-alone FastCGI version of PHP
php5-gd-5.2.6-no_x11 image manipulation extensions for php5
php5-mbstring-5.2.6 multibyte characters extensions for php5
php5-mcrypt-5.2.6 mcrypt encryption/decryption extensions for php5
php5-mysql-5.2.6  mysql database access extensions for php5
phpMyAdmin-2.11.7.1 tool to handle the administration of MySQL over the web
png-1.2.28        library for manipulating PNG images
t1lib-5.1.0p1     Type 1 rasterizer library for UNIX/X11
wget-1.11.3       retrieve files from the web via HTTP, HTTPS and FTP
#
```

可以看到，所需的软件包和依赖的包都安装好了！

四、Nginx 配置

1、Nginx 配置文件的修改

修改 nginx 的默认配置文件：

```
vi /etc/nginx/nginx.conf
```

按下面的内容修改 Nginx 的配置文件。蓝色表示需要手动修改的内容，红色表示增加的内容：

```
#user nobody;

# 指定子进程数，酌情修改
worker_processes 2;

#error_log logs/error.log;
#error_log logs/error.log notice;
#error_log logs/error.log info;

#pid      logs/nginx.pid;

# 最多可打开文件数
worker_rlimit_nofile 8196;

events {
    # 最大并发数
    worker_connections 1024;
}

http {
    include      mime.types;
    default_type application/octet-stream;

    # 关掉错误日志
    error_log     /dev/null crit;
    # 如果需要错误日志，就用下面这行替换上面这行
    #error_log    /var/log/nginx/error.log notice;

    # 定义日志格式，对日志使用缓存，避免频繁的磁盘 I/O 操作
    access_log    /var/log/nginx/access.log combined buffer=1m;

    sendfile      on;
    tcp_nopush    on;
    tcp_nodelay    on;

    keepalive_timeout 10;
```



```
# 对静态文件和可压缩文件启用压缩，以节约网络带宽，提高访问速度
gzip      on;
gzip_min_length 1k;
gzip_buffers 4 8k;
gzip_http_version 1.1;
gzip_comp_level 3;
gzip_types  text/html text/css text/xml text/plain application/x-javascript application/
xml application/pdf application/x-perl application/x-tcl application/msword
application/rtf application/vnd.ms-excel application/vnd.ms-powerpoint
application/vnd.wap.xhtml+xml image/x-ms-bmp;
gzip_disable  "MSIE [1-6] \.";
gzip_vary     on;

# 定义输出缓存大小
output_buffers 4 32k;

# 最大允许可上传文件大小
client_max_body_size 20m;

# 定义一个叫 “myzone” 的记录区，总容量为 10M
# 和下面的 limit_conn 一起限制单个 IP 的并发连接数为 10
limit_zone myzone $binary_remote_addr 10m;

server {
    listen 80;
    server_name localhost;

    location / {
        root /var/nginx/html;
        index index.php index.html index.htm;
        limit_conn myzone 10;
    }

    error_page 500 502 503 504 /50x.html;

    location = /50x.html {
        root /var/nginx/html;
    }

    location ~ \.php$ {
        root html;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
```

```
    fastcgi_param SCRIPT_FILENAME /var/nginx/html$fastcgi_script_name;
    include    fastcgi_params;
}

# 在浏览器本地暂存图片和静态文件，不记录日志，以节约机器资源
location ~* \.(gif|png|jpg|jpeg|bmp|css|js|swf)$
{
    root        /var/nginx/html;
    access_log  off;
    expires    max;
}

# 在浏览器中输入 http://xxx.xxx.xxx/status 可以看到 Nginx 的运行信息
# 需要密码认证，不记录日志，限制 IP 访问
location ~ /status
{
    auth_basic "O.N.M.P.";
    auth_basic_user_file password;
    stub_status on;
    access_log  off;
    allow 192.168.0.0/24;
    deny all;
}
}

}
```

其他的部分请酌情修改。

运行下面的命令生成查看 Nginx 运行状态的密码文件：

```
htpasswd -c /etc/nginx/password webadmin
```

按提示输入两遍密码即可。

在查看 status 的时候，输入用户名 webadmin（见上面这行）和密码就能够看到 Nginx 的运行数据了。

Nginx 能够流行和它的高负载能力是分不开的，在追求性能表现的场合，推荐使用 Nginx+PHP-fastcgi 的组合以获得强健的性能表现。而对于那些重视安全性的场合来说，可能 OpenBSD 内核集成的 Apache 更合适。OpenBSD 下搭建 Apache、MySQL、PHP 环境的详细内容请参见我的另一篇博文，地址在下面：

http://blog.chinaunix.net/u2/81136/showart_1860332.html

当然，你也可以利用 Nginx 内置的负载均衡功能，在前端分配访问流量，后端由 Apache 来运行 PHP 环境。Nginx 负载均衡的配置可以去 Nginx 的主页参看相关内容，地址：

<http://wiki.nginx.org/Main>。

下面为一个 Nginx 负载均衡的示例：

```
http {  
    upstream myproject {  
        ip_hash;  
        server 192.168.1.1:80;  
        server 192.168.1.2:80;  
        server 192.168.1.3:80;  
        server 192.168.1.4:80;  
    }  
  
    server {  
        listen 80;  
        server_name www.domain.com;  
        location / {  
            proxy_pass http://myproject;  
        }  
    }  
}
```

网络拓扑示意图如下：



II、Nginx 日志截断

OpenBSD 默认每天 00:00 会执行/etc/daily.local 脚本中的内容，我们只需要把 Nginx 日志截断的命令加入到这个文件中即可。

vi /etc/daily.local

加入下面的内容：

```
#!/bin/sh
# 对 Nginx 日志进行截断和压缩，以节约 log 分区空间
# 在张宴的基础上修改，感谢！
mkdir -p /var/log/nginx/${date -d "yesterday" +%Y}/${date -d "yesterday" +%m}/
mv /var/log/nginx/access.log /var/log/nginx/${date -d "yesterday" +%Y}/${date -d "yesterday" +%m}/access.${date -d "yesterday" +%Y%m%d}.log
kill -USR1 `cat /var/run/nginx.pid`
sleep 1
gzip /var/log/nginx/${date -d "yesterday" +%Y}/${date -d "yesterday" +%m}/access.${date -d "yesterday" +%Y%m%d}.log
```

保存退出，为/etc/daily.local 加上执行权限：

chmod 0755 /etc/daily.local

其他需要每天运行的命令也都可以加在这个脚本里面，各位自行处理。

五、MySQL 与 phpMyAdmin 的配置

I、MySQL 的配置

安全起见，MySQL 需要运行在自己的 daemon 下：

vi /etc/login.conf

跳到最后，加入 MySQL 所需的修改：

```
mysql:\
:openfiles-cur=2048:\
:openfiles-max=4096:\
:tc=daemon:
```

使修改生效：

cap_mkdb /etc/login.conf

初始化 MySQL 数据库：

/usr/local/bin/mysql_install_db

MySQL 自带了几个配置文件，在/usr/local/share/mysql 目录中，可以拷贝为/etc/my.cnf 使用。关于 MySQL 的配置和优化等内容，可以自行 Google，这里不再赘述。

MySQL 安全须知

1、不要用 root 身份运行数据库或 PHP 应用。并且，最好是数据库名和数据库用户名不同，以提高安全性。

例如，域名为 example.net，则数据库名设为 abcxyz，数据库用户名设为 xyzabc。总之，关联度越低越好，数据库名和数据库用户名越复杂越难猜测越好。

2、控制权限的分配，PHP 应用仅给予必要的权限。

例如，安装/升级 Discuz!和 PHPWind 论坛程序，仅需要下图所示的权限即可：



按数据库指定权限 (全选 / 全部不选)

注意: MySQL 权限名称会以英文显示

数据	结构	管理
<input checked="" type="checkbox"/> SELECT	<input checked="" type="checkbox"/> CREATE	<input type="checkbox"/> GRANT
<input checked="" type="checkbox"/> INSERT	<input checked="" type="checkbox"/> ALTER	<input type="checkbox"/> LOCK TABLES
<input checked="" type="checkbox"/> UPDATE	<input type="checkbox"/> INDEX	<input type="checkbox"/> REFERENCES
<input checked="" type="checkbox"/> DELETE	<input checked="" type="checkbox"/> DROP	
	<input type="checkbox"/> CREATE TEMPORARY TABLES	
	<input type="checkbox"/> CREATE VIEW	
	<input type="checkbox"/> SHOW VIEW	
	<input type="checkbox"/> CREATE ROUTINE	
	<input type="checkbox"/> ALTER ROUTINE	
	<input type="checkbox"/> EXECUTE	

执行

在安装/升级完毕后，还可以把 CREATE、ALTER、DROP 权限去掉，不会影响论坛程序的运行，而且提高了安全性！

II、phpMyAdmin 的配置

由于 OpenBSD 中的 phpMyAdmin 默认是安装在/var/www/phpMyAdmin 目录中的，直接使用 Nginx 是无法访问的，我们需要把它拷贝到 Nginx 目录下，这样就可以通过浏览器来管理 MySQL 数据库了。

mkdir /var/nginx/html/pma/


```
cp -rf /var/www/phpMyAdmin/* /var/nginx/html/pma/
```

修改 phpMyAdmin 的配置文件，使之可用。

```
vi +17 /var/nginx/html/pma/config.inc.php
```

将下面这行修改成：

```
$cfg['blowfish_secret'] = 'a'; /* YOU MUST FILL IN THIS FOR COOKIE AUTH!*/
```

（仅是加入了字母 a 而已）

保存退出。

小提示：

1、上面新建了/var/nginx/html/pma 目录来保存 phpMyAdmin 的文件，目录名很简单。你可以用一个更复杂的目录名来代替，例如：pma2YAY5jRpFfLXQVm 这样的目录名，以防止黑客利用字典攻击等方法渗透你的 phpMyAdmin！

2、你还可以在 Nginx 的配置文件中，将/var/nginx/html/pma 配置成某个虚拟主机的根目录，并限定可以访问这个虚拟主机的 IP。这样，就可以在很大程度上提高数据库的安全性了！一个配置示例在下面：

```
location / {  
    allow 192.168.0.0/24;  
    deny all;  
}
```

3、假如平时只是偶尔用到 phpMyAdmin，你还可以在用完后删除/var/nginx/html/pma 目录，需要用的时候再拷贝过去。

4、你还可以将以上方法结合起来灵活使用，既建立一个无法被猜测的目录名，又限制 IP 访问，这样你的系统被黑的机会就会小了很多！

六、强化 PHP 的安全

OpenBSD 软件包中的 PHP 已经自带了 suhosin 这个补丁，可以在很大程度上提高 PHP 脚本的安全。本节主要讨论在 php.ini 文件中进行相关的设置，进一步提升安全性。具体来说，就是禁用某些危险函数和启用 PHP 安全模式。

偶一般是把对 PHP 的所有修改都放在一个单独的文件中进行，包括对 PHP 参数以及扩展模块的修改，都放在这个文件中一并处理，这样查找、修改和管理会方便许多：

```
vi /var/www/conf/php5/addphp.ini
```

加入下面的内容：

；禁止动态加载模块

enable_dl = Off

；隐藏 PHP 信息

expose_php = Off

；限定可访问目录

open_basedir = /var/nginx/html/

；设定 session 暂存目录

session.save_path=/var/nginx/sesstmp

；设定 PHP 上传文件的临时目录

upload_tmp_dir=/var/nginx/tmp

；禁用危险函数（注意下面的内容应该是一行，编排的原因分成了多行）

disable_functions = phpinfo,com,shell,exec,system,passthru,error_log,
stream_socket_server,putenv,ini_alter,ini_restore,ini_set,dl,openlog,
syslog,readlink,symlink,link,leak,fsockopen,pfsockopen,proc_open,
popepassthru,escapeshellcmd,escapeshellarg,chroot,scandir,chgrp,chown,
shell_exec,proc_get_status,popen,shmop_close,shmop_delete,shmop_open,
shmop_read,shmop_size,shmop_write

；启用 PHP 的安全模式

；PHP 在安全模式下运行是用性能换安全。据简单测试，性能下降到 50%左右，各位请酌情使用

；启用安全模式后，某些程序可能受到影响。例如，Discuz!将无法上传附件

safe_mode = On

；pecl-APC 只使用 16M 的共享内存用以加速 PHP 程序的运行

apc.shm_size=16M

保存退出。

七、强化 SSH 的安全

1、SSH 配置文件的修改

vi /etc/ssh/sshd_config

跳到最后，加入下面的部分：

使用高位端口，防止黑客扫描 22 端口。可选范围 1024~65535，推荐 32768~65535。

Port 58937

登录时间控制在 30 秒内

LoginGraceTime 30

不允许 root 远程直接登录

PermitRootLogin no

StrictModes yes

最多允许三次错误

MaxAuthTries 3

最多允许三个 SSH 线程

MaxSessions 3

使用 SSH 协议 2

Protocol 2

不使用密码认证

PasswordAuthentication no

使用 KEY 的方式认证

PubkeyAuthentication yes

KEY 文件存放位置

AuthorizedKeysFile .ssh/authorized_keys

II、使用 KEY 进行验证

关于使用 PuTTY 进行 KEY 验证的方法，请参见偶的另一篇博文，地址：

http://blog.chinaunix.net/u2/81136/showart_1860332.html

III、按需启动 SSH

我们只在必要的时候启动 sshd 服务，用完即停止，不给黑客扫描的机会。输入下面的命令：

crontab -e

跳到最后，输入下面的内容（中间的空白部分为 Tab）：

每天的 10:30 启动 sshd 服务

30 10 * * * /usr/sbin/sshd

```
# 10:35 即停止 sshd 服务，也就是说，你只有五分钟的时间可以登录进系统
35 10 * * * kill `cat /var/run/sshd.pid`
```

保存退出。

上面的时间请酌情修改。sshd 服务启动的时间尽量不要太长，5~10 分钟应该够了！

IV、一点安全小常识

- 1、私钥请一定保存在安全的地方，不要保存在本地计算机上，并加上足够强度的“保护码”；
- 2、本地计算机不要保存远程主机的“指纹”，用一次确认一次（远程主机的指纹保存在注册表的 HKEY_CURRENT_USER\Software\SimonTatham\PuTTY 项下，可以手动删除）；
- 3、不要使用 DSA 密钥，据说有漏洞的；
- 4、不要在公用计算机上连接远程主机，有键盘记录器、木马什么的就麻烦了；
- 5、多个远程主机不要使用相同的公钥，最好是使用各自的公钥、私钥；
- 6、少用，最好是不用无线网络，加密强度太差，容易被破解；
- 7、M\$系统下的病毒、木马太多了，推荐客户端转换到 Linux/UNIX 环境，安全性更高，而且仍然有 PuTTY 及工具可用；
- 8、注意清除本地机器上不必要保留的文件（例如私钥、公钥、主机 RSA 指纹、C:\Documents and Settings\Administrator 目录下的文件等等）。

八、启用 Packet Filter 防火墙

Packet Filter 是非常优秀的包过滤防火墙，OpenBSD 核心已经集成了 Packet Filter 防火墙，不过默认并没有启用，下面我们来启用 Packet Filter 的强大功能！

```
vi /etc/pf.conf
```

跳到最后，加入下面的内容：

```
# 宏定义
# 请把下面的 fxp0 换成你自己用的外网网卡，不知道的可以输入 ipconfig 查看
ext_if="fxp0"

# 指定可以使用 SSH 登录的 IP，支持 CIDR
admin_add="192.168.0.0/24"
```

```
# 指定 SSH 端口。
# 注意，如果在/etc/ssh/sshd_config 文件中更改了 SSH 端口号，这里的也需要同样修改。否则连
# 不上了不要怪偶没有提醒！
ssh_port="58937"

# 维持一个持久的表，里面存放的是对本机发动 DDoS 攻击的 IP
table <badhosts> persist

# 选项设定
set require-order yes
set block-policy drop
set optimization aggressive
set loginterface none
set skip on lo0

# TCP 参数设定
set timeout {interval 3,frag 10}
set timeout {tcp.first 10,tcp.opening 2,tcp.established 600,tcp.closing 20,tcp.finwait
10,tcp.closed 10}

# UDP、ICMP 及其它参数设定
set timeout {udp.first 20,udp.single 10,udp.multiple 10}
set timeout {icmp.first 10,icmp.error 5}
set timeout {other.first 20,other.single 10,other.multiple 20}
set timeout {adaptive.start 0,adaptive.end 0}

# 允许最多有 65536 个连接
set limit { states 65535, frags 200, src-nodes 65536, tables 65536, table-entries 1048576 }

# 包整形
scrub in all
scrub out all

# 阻止所有不匹配的包和从 DDoS 主机来的包
block quick from <badhosts>
block return
block in all
block out all

# 防止 IP 欺骗
antispoof quick for {lo0,$ext_if}

# 允许本机访问其他机器
```


pass out quick on \$ext_if inet from \$ext_if to any flags S/SA keep state

允许 IPv4 地址的客户机访问本地 80 (www) 端口，发起过快连接 (DDoS) 的主机加入阻止列表。
注意是一行，下同

```
pass in quick on $ext_if inet proto tcp from any to $ext_if port 80 flags S/SA synproxy state
(source-track rule,max-src-nodes 200,max-src-states 100,max-src-conn 100,max-src-conn-
rate 1000/10,overload <badhosts> flush global )
```

允许 IPv6 地址的客户机访问本地 80 (www) 端口，发起过快连接 (DDoS) 的主机加入阻止列表

```
pass in quick on $ext_if inet6 proto tcp from any to $ext_if port 80 flags S/SA synproxy state
(source-track rule,max-src-nodes 200,max-src-states 100,max-src-conn 100,max-src-conn-
rate 1000/10,overload <badhosts> flush global )
```

允许管理 IP 远程连接本机 SSH 端口

```
pass in quick on $ext_if inet proto tcp from $admin_add to $ext_if port $ssh_port flags S/SA
synproxy state
```

保存退出。

修改系统配置，使得开机启用 PF 防火墙：

```
vi /etc/rc.conf.local
```

跳到最后，加入下面这行：

```
pf=YES
```

保存退出。

九、系统启动脚本的修改

修改系统启动脚本，使得 MySQL、PHP(fastcgi)、Nginx 可以在系统启动的时候自动启动，免去手动启动的麻烦。

```
vi /etc/rc.local
```

跳到最后，加入下面的内容：

```
# 启动时校时。这行也可以加在/etc/daily.local 文件的最前面，每天零点自动校时
rdate -n 210.72.145.44
# Start MySQL
if [ -x /usr/local/bin/mysqld_safe ] ; then
    echo -n 'Starting MySQL...'
    su -c mysql root -c '/usr/local/bin/mysqld_safe >/dev/null 2>&1 &'
```

```
    echo "DONE"
fi
# Start php-fastcgi
if [ -x /usr/local/bin/spawn-fcgi ] ; then
    echo -n 'Starting php-fastcgi...'
    /usr/local/bin/spawn-fcgi -a 127.0.0.1 -p 9000 -C 6 -u www -f /usr/local/bin/php-fastcgi > /
var/run/fcgi.pid
    echo "DONE"
fi
# Start nginx
if [ -x /usr/local/sbin/nginx ] ; then
    echo -n 'Starting nginx...'
    /usr/local/sbin/nginx
    echo "DONE"
fi
```

保存退出。

reboot 重启，启动后输入 top 看看，O.N.M.P.环境已经搭建好了！

```
load averages: 0.12, 0.10, 0.08
26 processes: 25 idle, 1 on processor
CPU states: 0.2% user, 0.0% nice, 0.0% system, 0.0% interrupt, 99.8% idle
Memory: Real: 54M/107M act/tot Free: 389M Swap: 0K/502M used/tot

  PID USERNAME PRI NICE  SIZE  RES  STATE  WAIT  TIME  CPU COMMAND
  4215 _mysql    2    0  276M   33M  sleep  poll   0:00  0.00% mysqld
16033 www       2    0   21M  6384K  idle   poll   0:00  0.00% php-fastcgi
  6750 www       2    0   21M  1600K  idle   poll   0:00  0.00% php-fastcgi
20362 www       2    0   21M  1588K  idle   poll   0:00  0.00% php-fastcgi
14668 www       2    0   21M  1588K  idle   poll   0:00  0.00% php-fastcgi
21492 www       2    0   21M  1588K  idle   poll   0:00  0.00% php-fastcgi
19544 www       2    0   21M  1588K  idle   poll   0:00  0.00% php-fastcgi
28668 www       2    0   21M  1588K  idle   poll   0:00  0.00% php-fastcgi
  9233 _nginx     2    0   11M   11M  idle   kqread 0:00  0.00% nginx
31257 root        2    0 3348K 2372K  sleep  select 0:00  0.00% sshd
32730 root        2    0  968K 1452K  sleep  select 0:00  0.00% sendmail
19445 root       18    0  816K  948K  idle   pause  0:00  0.00% nginx
```

系统启动后，还要把 MySQL 自带的测试数据库和匿名用户删除，防止被黑客利用。默认情况下，新安装的 MySQL 数据库，root 密码为空！同样需要第一时间进行设置，此处不再赘叙。

写在最后

其实，系统的安全是个整体工程，并不是用上了 OpenBSD 这个最安全的操作系统就算万事 OK，还有很多事情要做。

更多的是需要在日常工作中积累经验，多分析系统整体的运行情况，多关注网络安全方面的内容，这样才能尽可能的打造安全的运维环境。本手册只是起到抛砖引玉的作用，希望能吸引更多的人来关注系统安全，希望有更多的人能用上 OpenBSD 这个主动安全的操作系统，希望能推动 OpenBSD 在国内的更多普及。如此，则幸甚！

网友热评

热点技术评论

[一个面试题——文件内容按行排序问题](#)

[通过 ssh 远程执行命令问题](#)

[大家看看这条 SQL 还有优化的余地么？](#)

[DirectAdmin 做 VPS，如何架设 JSP](#)

[为何使用 cout 不能输出 volatile 变量的地址？](#)

[TCP 发送端口如何选择？](#)

[Haskell 中的语法糖](#)

[nbu 备份 sun cluster 环境的 oracle 的特殊设置](#)

[OpenBSD4.4 搭建“CS 起源”服务器](#)

[数组这样操作为什么不越界](#)

[删除特定文本的 shell](#)

[OpenVPN+radius+linux 怪异的服务器掉线问题](#)

[关于 sort 或者 awk 合并文件](#)

[Linux 普通用户提权 root SHELL 版首发](#)

[RHEL 5 重启后网卡设备名变换](#)

[Bash 强大的 History 命令](#)

[模块编程](#)

[十万火急的关于修改 root 密码的问题](#)

[IP_conntrack_max 问题](#)

[linux 下磁盘 lvm 挂在显示容量问题](#)

[嵌入式 linux 下 printf printk](#)

[求救 Redhat Linux 系统重启，盘阵找不到！](#)

[这两个函数哪个速度快？](#)

[当前目录生成 XML 文件](#)

[gcc 认识 pow](#)

[epoll_ctl 出现 Invalid argument](#)

热点新闻评论

[十大原因宣判 Windows 死亡 Linux 终将胜利](#)

[读完<计算机程序设计艺术>4 卷后,年薪多少？](#)

[终于发现 unix 的 bt 之处了](#)

[Rackable2500 万美元收购 SGI](#)

[前天去 IBM 的一个笔试题](#)

[HW 怎么不收购 SUN？](#)

[《算法导论》好看吗？](#)

[独家：未来五年程序员应当具备的十项技能](#)

[月薪一万在上海的生活 IT 真实版](#)

[我不得不炒大学生的理由](#)

[SUN 又参加性能测试了](#)

[SUN 和 Oracle 成交](#)

[软件测试工程师前景怎样？](#)

[方舟子打假唐骏 第一职业经理人是假博士？](#)

[我倡导一个 linux 推广计划——linux 西时工程](#)

[Linux 爆本地提权漏洞 请立即更新 udev 程序](#)

[RHCE 已经开始坠落](#)

[诚邀参加 SUSE Linux Enterprise 11 技术大会](#)

[喜欢 linux，考研应该考哪个方向？](#)

[红帽面试的问题](#)

[大家欣赏一下 linux 红屏的样子吧！](#)

[Debian 支持 FreeBSD Kernel](#)

[OS 市场应该是 MacOS 居大才对！](#)

[从 java 转入手机软件开发，如何入手](#)

[1.4 亿 IC 卡安全有隐患：掌握技术可修改钱数](#)

[关于职业生涯方面的困惑,请大家来讨论](#)