# Linux运维趋势

第十五期

2011年12月号 关键字：Syslog, Eucalyptus, OpenStack, 智能DNS

## 虚拟化管理软件应用与选型

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

# 目录

人物・People

003 专访章文嵩博士：做系统要先了解业务的需求

交流・Interact

005 红帽开发者：准备扔掉你的syslog吧！

八卦・News

007 RHEL、CentOS、openSUSE纷纷升级

专题・Special

009 虚拟化管理软件比较 －－ 综合篇

011 使用Eucalyptus打造自己的云测试平台

013 以公司实际应用讲解OpenStack到底是什么

015 为 OpenNebula 制作 Ubuntu 镜像

017 使用CONVIRT管理基于KVM的虚拟机

技巧・工具・脚本・DBA

019 MySQL主从配置的一些总结

021 智能DNS(Bind dlz)在企业中的应用

023 针对Linux集群的高级监控工具sinfo概述

025 系统管理员必须知道的PHP安全实践

Linux 运维趋势

出版方：51CTO 系统频道 (北京无忧创想信息技术有限公司)
杂志主编：杨赛
联系方法：yangsai@51cto.com 010－68476606 (分机 8035)
出版日期：2011 年 12 月 16 日
每月第 2 个星期五出版
订阅：http://os.51cto.com/art/201011/233915.htm

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

# 专访章文嵩博士：做系统要先了解业务的需求

采访整理/lazycai

在今年的 O'Reilly Velocity China 2011 会议上，51CTO 编辑有幸采访到了 LVS 创始人、目前就职淘宝的章文嵩博士，请他对淘宝这几年核心系统优化团队的工作，以及他个人所关注的方向进行了介绍。

**51CTO：您到淘宝两年多来，一开始主要是 LVS 和 HAproxy 的大规模普及，之后做了很多 CDN 系统的改良工作，今年又推出 GreenCompute 项目。能介绍一下这几年工作的整体规划思路吗？**

章文嵩：优化其实是个一直持续的工作。最早，淘宝的 CDN 用的是商用的调度负载均衡器 Citrix NetScaler，这是当时业界最好的负载均衡器。但是淘宝因为规模越来越大，容量也很大，有很多针对小图片方面的需求，用 Citrix 就遇到很多问题。比如流量方面，小图片造成的请求特别多，但是万兆网卡的流量只能到 3G，一旦流量超过 3G，哪怕只是到了 4G，系统就会崩溃。这不仅是前面的负载均衡，在后端的缓存服务器上也是。在 CDN 系统中，图片处理的挑战最大，相比视频那种连续的数据，图片这种很小的、比较离散的数据，对硬盘的访问要求很高。淘宝在 2008 年底到 2009 年初左右，曾经一度全用 SSD，这样下来一个 CDN 的节点造价就会比较高，要花掉两百万左右的钱，还要再加上商用的负载均衡器。而且刚才也说了，一个 NetScaler 流量只能提供到 3G，两台加起来也就只有 6G，而且两个 NetScaler 还不敢用心跳线，因为它虽然支持这个功能，但是万一有一台坏掉，6G 的流量完全转移到另一台，那结果肯定会崩溃掉。所以这就是商用系统的问题：在特别的负载情况下，它是不适用的。

**“图片这种很小的、比较离散的数据，对硬盘的访问要求很高。”**

所以我们就开始逐步改造。用 LVS+HAproxy，在硬件高配的情况下，一个节点跑到 100G 都没问题。成本方面，后端肯定不能无限制的花钱，我们就开始用混合存储，SATA、SAS 和 SSD 都有。经过优化之后，效果做到跟 SSD 差不多，存储空间更大，命中率更高，像我们现在有些高的，命中率可以做到 98%。那么现在我们一个最低标准的 CDN 节点，流量在 10G，成本已经优化到 50 万；如果低功耗的话，还可以进一步优化到 37 万。这个项目我们以后还会持续优化，因为优化无止尽嘛，我们追求的目标是更好的用户体验，更短的响应时间，同时还要花更少的钱。

响应时间是我们最关注的指标，因为它直接影响到用户的体验。其实淘宝目前在图片的优化方面做得算还不错，好比今年双十一的时候，我们最高跑到了 820G 的瞬间流量，这在世界上可能也是一个记录了。其实你现在打开一个淘宝的网页，图片加载的速度已经挺快了，目前我们图片请求的平均响应时间已经在 10ms 以下；但我们还会进一步挖掘，看能不能到 9ms 以下或 8ms 以下，让它加载的更快。虽然说，越往下面挖掘，难度会越来越大，但是这也是值得去做的，因为我们的规模很大。而且为了让用户体验好，我们多花点钱也是应该的。

另一方面，网络传输这一块也往往是瓶颈。一个图片请求发过来，硬盘

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

的处理时间算 10ms，但是在网络传输方面，CDN 部署的好的情况，可能会占用 20ms，万一网络有问题的时候就可能到 70、80ms。所以网络协议的优化方面，可以挖掘的空间可能会更大一些。

**“做系统，或者做基础架构设计，最重要的是先去了解业务的需求。”**

**51CTO：您这次在 Velocity 大会上主要讲的是 GreenCompute 的项目。这个项目是完全自己做的，还是通过借鉴一些案例来做呢？**

章文嵩：GreenCompute 项目是绿色计算嘛，我们最早做的项目就是低功耗服务器。

2008 年的时候，我自己在做一个 ARM 下载盒的项目，400MHz，装的 Linux 系统，空闲时候的功耗只有 1W，接外部大容量硬盘的情况，满负荷跑起来也就是 9W。当时做这个的时候，因为考虑到互联网上的很多应用都是数据密集型的，这些应用主要的任务都是硬盘读写和网络传输，实际的计算并不复杂，用不到多少 CPU 资源。相比一个高性能的服务器放在那里跑，空闲情况都有 200W 的功耗， 这种应用用低功耗服务器来跑是非常划算的。而低功耗服务器方面，选择不同的处理器，功耗和效果都会有所不同，主要考虑的是一个平衡点的问题，跟你的应用性质有很密切的关系。

国外大学对这方面的研究蛮多的。2009 年 5 月份，HotOS（美国一个操作系统热点主题的会议）上有一个 CMU（卡内基梅隆大学）的同仁做的一个 FAWN Cluster，他就是拿 AMD 那个 500MHz 的 Geode 处理器，加上我们照相机用的那种 4G 的 CF 卡，加上 256MB 的内存，这样下来一片的整体功耗只有不到 4W，这个在当时 MIT 的 Technology Review 上报道过。另外，美国加州大学圣地亚哥分校（UCSD）也有对 Gordan 架构的研究，那个是用 Atom 处理器做的低功耗服务器。每一家设计的都是针对自己独特的应用，效果都不错，对我们业界来说是很好的参考。

2009 年底，我们立项要做低功耗的项目，选择了当时我们掌控的比较好的 CDN 系统。CDN 因为对数据的安全性要求不是太高，毕竟上面都是缓存，数据丢了就丢了，全局系统可以随时把坏掉的节点切走，对用户的影响低，所以这个低功耗服务器就专门针对 CDN 应用进行定制。这中间经历了很多事情，整个过程并不很顺利，最终成功弄出来，也要感谢很多参与的厂商，像威盛，Intel，还有超微。总之我们最终还是把这个基于 Atom 的服务器用起来了，而且性能优化效果目前跟其他的 Intel 处理器相比，效果是最好的。

**51CTO：您对底层系统工程师有什么建议吗？比如未来三年需要掌握哪些技术之类的。**

章文嵩：说实话，我没有那么好的预见性啦。我自己觉得，做系统，或者做基础架构设计，最重要的是先去了解业务的需求。整体的需求是怎么样？访问的特点是怎么样？很多时候要去做取舍。我们设计架构的，或者设计系统的，或者哪怕是优化，很多时候都是做取舍的过程。对这个应用来说，什么是最重要的？我要抓住最重要的，而那些不太重要的，我就简单的做，或者不怎么做。把最主要的抓住，就能够针对每一个应用把性能做到极致。尤其是规模特别大的系统，哪怕我性能只是优化了 1%，那么对于 1 万台机器的环境，这就意味着可以节约 100 台。规模足够大的环境，这种优化值得去很深入的做。你的规模有多大，决定了你的研发成本是否值得——这个过程本身也是要做取舍的。其实我们在生活上也是一样的：什么是最重要的，我就会花最大的力气去解决。

由于篇幅所限，本文为删节版，完整内容见原文：

http://os.51cto.com/art/201112/306707.htm

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

# 红帽开发者：准备扔掉你的syslog吧！

文/Srinath
编译/布加迪

多年以来，系统管理员们和运维们在怀疑服务器遭受攻击或出现问题的时候，肯定是要先检查 syslog 的。不过在最近，红帽的两位开发人员提议采用一种新的基于二进制数的工具“The Journal”，这个工具可能将在 Fedora 17 中取代 syslog。

两位开发人员名叫 Lennart Poettering 和 Kay Sievers。他们的建议是：现在的 syslog 已经是 30 年的老古董，不仅效率低下，很容易被误读和被改动。

这在很大程度上归咎于系统日志不拘形式的性质：只要是 Linux 系统上的应用程序或守护程序发送的文本字符串，不管采用什么样的形式，系统日志基本上都照收不误。于是，某个守护程序可能会以某一种方式发送关于事件的信息，而另一个守护程序可能会以全然不同的方式来发送事件信息；这样一来，解析信息其含义的任务就扔给了阅读日志的人。自动化的日志分析工具在这方面有所帮助，但是 Poettering 和 Sievers 在关于 The Journal 的详细描述中写道：

**“kernel.org入侵事件涉及的就是黑客操纵日志文件；要发现这种攻击行为，完全靠运气。”**

“记入日志的数据其形式非常随便。自动化的日志分析工具需要解析人类语言字符串，以便：

1）识别消息类型

2）从中解析相关参数

这就导致了令人讨厌的正则表达式，而且经常需要跟在上游的开发人员屁股后面，因为这些开发人员可能会在新版本的软件中调整人类语言的日志字符串。为了不破坏用户所用的正则表达式，所有日志消息变成了其相对应的服务的二进制文字版界面（ABI），而这通常不是开发人员的本意。”

这两位开发人员重点指出了当前的系统日志体系存在的 14 个问题，而上面这个只是其中之一。其他问题包括如下：

◆ syslog 的数据没有经过验证。

◆ syslog 仅仅是 Linux 上众多的日志系统之一。

◆根本就没有针对 syslog 的访问控制机制。

◆只是在固定的间隔时间对磁盘使用实行了限制，导致系统很容易受到 DDoS 攻击。

Poettering 和 Sievers 重点指出了 syslog 体系存在的一个大家非常关注的问题：

“比如说，最近热议的 kernel.org 入侵事件涉及的就是黑客操纵日志文件；要发现这种攻击行为，完全靠运气。”

考虑到这些因素，Sievers 和 Poettering 提议采用 The Journal 守护程序，该守护程序将来自系统日志的事件以二进制数、而不是文本的形式来存储

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

数据，将数据作为包含散列以增强安全性的键－值对列表来存储。

**“许多灵感其实源自Git版本控制系统的键/值、散列和元数据这些概念。”**

这并不是这两位开发者头一回提议对 Linux 系统的基础架构作出如此全面的改变。Poettering 不但是 PulseAudio 声音服务器的开发者，还开发了取代 Linux 上 System V init 守护程序的 systemd 守护程序。Sievers 最近成了 Fedora 项目团队的一名成员，他提议：需要时，将所有可执行文件移入到 /usr/bin 目录，将它们的库移入到 /usr/lib 或 /usr/lib64。

实现了这种二进制数后，The Journal 守护程序就能够为每个系统事件添加元数据，比如进程编号和发送者名称、用户和用户组编号以及其他关键的系统数据。

“受 udev 事件的启发，The Journal 条目酷似环境块。许多键 / 值字段由换行符分隔，使用大写字母的变量名。与 udev 设备事件和实际环境块相比较，有一大区别是：虽然关注的重心绝对放在 ASCII 格式化字符串上，但是作为值的二进制斑点 (binary blob，装入到开源操作系统内核里面的一种对象文件，目前尚未开源) 也得到支持——这种对象文件可以用来添加二进制元数据，比如 ATA SMART 健康状况数据、SCSI 感知数据、核心转储数据或固件转储数据。生成 The Journal 条目的代码，想为添加多少字段，就可以添加多少。字段可以是常用字段，也可以是针对特定服务 / 子系统 / 驱动程序的字符。”

如果说开发人员觉得这一切听起来有点耳熟，那么不妨直说吧：Poettering 和 Sievers 在这方面的许多灵感其实源自 Git 版本控制系统的键 / 值、散列和元数据这些概念。

实施 The Journal 不但会让 Linux 系统变得更安全，其发明者还希望通过统一 Linux 机器上的所有日志系统，为数据高效地重新建立结构，可以实际减少日志系统在 Linux 上占用的资源。

“其设计方式如下：日志数据只添加在末尾，头里面的一些元数据变化可以引用新添加的日志数据。字段在日志文件中作为一个个对象来存储，然后可供有需要的所有条目来引用。这大大节省了磁盘空间，因为日志条目通常高度重复 (想一想每个本地消息都会含有同样的 _HOSTNAME= 和 _MACHINE_ID= 字段)。数据字段经过了压缩，目的是为了节省磁盘空间。最终结果就是，虽然 The Journal 记入日志的元数据要比经典系统日志记入的多得多，但是占用的磁盘空间并没有立马体现这一点。”

然而，不是每个人都因这一提议而激动万分。许多人在最先刊登这项提案的 LWN 上发表了反对意见，他们为简单的基于文本的系统将换成依赖 The Journal 这一种工具的二进制数据格式而悲痛，而这个工具将仅在 systemd 守护程序中存在。有几个人在 The Journal 的 FAQ 留言道：

“日志文件格式会实现标准化吗？”

“目前，我们不想对格式进行标准化。只要我们觉得合适，就想随意改变格式。我们可能最终会将磁盘上数据格式记入文档，但是眼下，我们不想使用其他任何软件来直接读取、写入或操纵我们的日志文件。我们需要一个共享库和命令行工具才能访问。”

这在更广泛的 Linux 社区会引起怎样的反响？显然，Linux 在经历一些重大的革命性变化，剔除了 UNIX 的一些糟粕。在 Linux 不断前进的同时，这些变化会给它带来怎样的影响，让我们拭目以待吧。

原文：Linux syslog may be on way out

译文：http://os.51cto.com/art/201111/304174.htm

相关阅读：The Journal 详细解读

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

# RHEL、CentOS、openSUSE纷纷升级

——八卦，新闻与数字 2011.11 – 2011.12

【Apache】12 月份 Web 服务器最新数据：在全球 555,482,744 个调查网站中，使用 Apache Web 服务器的网站有 362,267,922，占全球市场份额的 65.22%；Nginx 由 8.5% 上升至 8.85%。

http://os.51cto.com/art/201112/307412.htm

【WebOS】惠普将向程序员公开 WebOS 源代码，把 WebOS 推向开放源代码社区。

http://os.51cto.com/art/201112/306831.htm

【RHEL】Red Hat 宣布发布 Red Hat Enterprise Linux 6.2。主要新特性包括：改进和增强存储和文件系统性能；支持 PCI–e 3.0 和 USB 3.0；支持多种新款 10 GbE 网络适配器和主机总线适配器，以及两用统合式网络适配器；简化配置和部署 FCoE 等等。

http://os.51cto.com/art/201112/306345.htm

【CentOS】CentOS 6.1 正式版发布了。

http://os.51cto.com/art/201112/306855.htm

【openSUSE】openSUSE 12.1 正式发布，主要更新包括 GNOME 3.2/KDE 4.7，Apper 软件管理工具，Oyranos KDE 色彩系统，管理 BTRFS 快照的图形化工具 Snapper，systemd 加速系统启动等。

http://os.51cto.com/art/201111/302601.htm

@ 朱佳文 _：从可管理性讲，支持 rpm 方式，一是编译安装时编译选项有个人喜好的因素，不好统一；二是生产环境不建议有编译环境。如果有统一的打包规范，包管理，包安装，就有很好的应用环境管理手段

http://weibo.com/1899267217/xzFIU91PE

@NinGoo：豆瓣有超过 200 员工，5000 万用户，60 台服务器，其中十几台离线计算，应用和数据库各二十几台，每月产生 1T 日志，使用 MooseFS，Infobright，Hadoop 做为基础设施

http://weibo.com/1645755853/xAUIzxGt9

@stvchu：" 硬盘是不可靠的，服务器是不可靠的，机房是不可靠的，网络是不可靠的，工程师是不可靠的，代码是不可靠的，政府是不可靠的。这些是分布式系统及其理论要解决的核心问题之一。"

http://weibo.com/1614209091/xALRy9Crx

@ 勇哥 Beston：淘宝 Web 服务器 Tengine 正式开源 ( 迟到的资讯 )
http://t.cn/SGkcxZ

http://weibo.com/1699370621/xAJJzq5jC

@Fenng：Linux.com 的一篇标题党文章，2011 年最重要的开源项目，名单：Hadoop、Git、Cassandra、LibreOffice、OpenStack、Nginx、jQuery、Node.js、Puppet、Linux。可作为技术选型的一个参考。不过，Cassandra 与 OpenStack 混在里面感觉很可疑。

http://weibo.com/1577826897/xBZbhwN1e

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

# 虚拟化管理软件应用与选型

【编者按】过去几年间，虚拟化领域的发展非常迅速。Linux 虚拟化领域的 VMware、Xen 都已经十分成熟，红帽的 KVM 也在大步跟进。

而随着 Amazon 云服务的流行，这种 IaaS/PaaS 的应用服务也因为其管理的便捷和资源高度可调节性而广受关注，国内的厂商和政府部门也纷纷展开各自的 IaaS 和 PaaS 建设工程，很多企业内部也在思考，究竟这种所谓的公有云和私有云，能够为自己带来什么好处。

无论是哪种服务，其中最关键的一个环节就是计算资源的统一分配管理。在开源领域，近几年出现了很多这方面的项目，比如最早和 Ubuntu 关系紧密的 Eucalyptus 项目，由 RackSpace 和 NASA 联合开源的 OpenStack 项目，美国阿贡国家实验室推出的 Nimbus 项目，由马德里的两位计算机系教授发布并开源的 OpenNebula 项目，由 XenMan 演变而来的 Convirt 项目等等。

在下面的专题中，我们将对这些虚拟化管理软件的选型进行一些简略的讲解，并对其中的几个较为成熟的项目进行介绍。

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

# 虚拟化管理软件比较 —— 综合篇

文/蒋清扬

目前市面上形形色色的虚拟化管理软件总数很多，这一系列文章所提及的几个软件仅仅其中的几个代表。

(1) 商务评估

从商务上进行软件选型，性价比通常是一个决定性的因素。在假定参与选型的软件全部满足技术要求的前提下，企业(机构) 需要考虑的因素包括软件的授权协议是否友好、许可证管理的难易程度、软件和服务的价格高低、运营团队在业界的声誉、开发者社区和用户社区的规模和活跃程度、商业与技术沟通的难易程度。

授权协议 / 许可证管理 — 以全部开放源代码为 10 分，部分开放源代码(例如以企业版的形式提供某些高级功能，或者以服务的形式提供特别版本的安装包和补丁)扣 1 分。商业版本需要在控制节点安装许可证不扣分，需要在所有计算节点安装许可证扣 1 分，许可证需要每年更新者扣 1 分。

价格指数 — 以全部功能免费使用为 10 分，以企业版的模式提供全部功能的软件，每台物理服务器每花费 500 美元扣 1 分。

运营团队 — 以运营团队的规模、背景、影响力评分，存在的主观因素较多。

社区因素 — 以开发者和用户社区的规模和活跃程度评分，存在的主观因素较多。

沟通交流 — 以个人与运营团队、开发者社区、用户社区之间的沟通顺畅程度评分，存在的主观因素较多。

| | 授权协议 | 价格指数 | 运营团队 | 社区因素 | 沟通交流 | 总分 |
|---|---|---|---|---|---|---|
| Eucalyptus | 9 | 8 | 9 | 9 | 10 | 45 |
| OpenStack | 10 | 10 | 8 | 8 | 7 | 43 |
| OpenNebula | 9 | 9 | 7 | 8 | 9 | 42 |
| OpenQRM | 9 | 8 | 6 | 7 | 8 | 37 |
| XenServer | 7 | 8 | 9 | 10 | 9 | 43 |
| Oracle VM | 9 | 7 | 7 | 6 | 7 | 36 |
| CloudStack | 9 | 8 | 7 | 6 | 7 | 37 |
| ConVirt | 9 | 8 | 8 | 9 | 10 | 44 |

(2) 功能评估

从功能上进行虚拟化管理软件选型，需要考虑的因素包括该软件所支持的虚拟化技术、安装配置的难易程度、开发和使用文档的详尽程度、所提供的功能是否全面以及用户界面是否直观友好、二次开发的难易程度、是否提供物理资源和虚拟资源的监控报表等等。

虚拟化技术支持 — 仅支持一种虚拟化技术为 6 分，每增加一种虚拟化技术加 1 分，10 分封顶。

安装配置 — 以按照官方文档进行安装配置的难易程度评分，存在的主观因素较多。

开发 / 使用文档 — 以官方所提供的开发与使用文档的详尽程度评分，文档详尽程度越高者得分越高。

功能与界面 — 综合评分，涵盖用户进行物理资源和虚拟资源管理、虚

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

拟机生命周期管理、访问虚拟机资源和存储资源的难易程度，用户界面的美观易用程度，以及综合用户体验。

二次开发 — 基础得分 6 分，提供与 Amazon EC2 相兼容的程序调用接口者加 3 分，提供二次开发接口但是与 Amazon EC2 不兼容者加 2 分。

监控报表 — 基础得分 6 分，依系统所提供监控与分析功能的详尽程度加分。

| | 虚拟化支持 | 安装配置 | 开发/文档 | 功能与界面 | 二次开发 | 监控报表 | 总分 |
|---|---|---|---|---|---|---|---|
| Eucalyptus | 8 | 8 | 9 | 4 | 9 (AWS) | 6 | 44 |
| OpenStack | 10 | 8 | 8 | 4 | 9 (AWS) | 6 | 45 |
| OpenNebula | 8 | 8 | 7 | 4 | 9 (AWS) | 6 | 42 |
| OpenQRM | 10 | 9 | 5 | 10 | 6 (OS) | 7 | 47 |
| XenServer | 6 | 10 | 10 | 10 | 8 (Plugin) | 9 | 53 |
| Oracle VM | 6 | 9 | 8 | 7 | 8 (WS) | 7 | 45 |
| CloudStack | 8 | 9 | 8 | 10 | 6 (OS) | 8 | 49 |
| ConVirt | 7 | 10 | 10 | 10 | 8 (API) | 10 | 55 |

(3) 综合评估

从商务上考虑，Eucalyptus 和 ConVirt 以微弱的优势领先于其他选项。Eucalyptus 是私有云管理平台的先行者。Ubuntu 10.04 选择捆绑 Eucalyptus 作为 UEC 的基础构架，使得 Ecualyptus 比其他的私有云管理平台拥有更多的用户和更加活跃的社区。此外，Ecualyptus 在中国国内有销售和技术支持人员，在沟通上比选择其他软件要更加容易。ConVirt 排名第二，根本原因在于其销售和技术支持团队与（潜在的）客户保持积极而有效的沟通。Citrix XenServer 仅仅与其他两个选项并列排名第三，输在其过于严苛的许可证管理政策。的确，要给 100 台以上的服务器单独安装许可证并且每年更新一次，可不是一件有意思的事情。

从功能上考虑，ConVirt 与 XenServer 遥遥领先于其他选项。虽然 ConVirt 仅仅支持 Xen 和 KVM 两种虚拟化技术，但是其安装配置相对简单，文档详尽、功能齐全、界面美观、是比较容易上手的虚拟化管理软件。更重要的是，ConVirt 的监控报表功能直观地展示了从数据中心到虚拟机的 CPU、内存利用情况，使得用户对整个数据中心的健康状况一目了然。同样，XenServer 虽然仅支持 Xen 一种虚拟化技术，但是在安装配置、操作文档、用户界面等方面都不亚于 ConVirt。如果用户对基于 Windows 的界面没有强烈的抵触情绪的话，XenServer 是比较值得考虑的一个选型。

综合如上考虑，对于希望利用虚拟化管理软件提高硬件资源利用率和虚拟化管理自动化程度的企业（机构）来说，建议使用 ConVirt 来管理企业（机构）的计算资源。如果网管人员不希望深入了解 Linux 操作系统，并且所管理的物理服务器数量有限的话，XenServer 也是一个不错的选择。ConVirt 的浏览器界面是开放源代码的，用户可以对其进行定制化，将自己所需要的其他功能添加到同一个用户界面中去。XenCenter 则提供了一种插件机制，用户可以通过插件的方式讲自己的功能集成到 XenCenter 中。

不过，你的基础设施是否需要与 Amazon EC2 相兼容呢？也就是说，你的用户是否需要使用他们用于访问和操作 Amazon EC2 的脚本和工具来访问你的计算资源呢？如果是这样的话，你可能需要在 Eucalyptus 和 OpenStack 之间作一个选择（CloudStack 和 OpenNebula 同样提供了与 Amazon EC2 兼容的操作接口，但是 CloudStack 在商务方面得分不高，OpenNebula 在功能方面得分不高）。Eucalyptus 的历史比 OpenStack 稍长，用户群比 OpenStack 要大，社区的活跃程度也比 OpenStack 要高。不过 OpenStack 的后台老板 NASA 比 Eucalyptus 要财大气粗，Ubuntu 11.04 也集成了 OpenStack 作为其 UEC 的基础构架之一，表明 OpenStack 已经得到了社区的重视和支持。总的来说，开放源代码的云构架，还是一个不断发展之中的新生事务。笔者只能够建议用户亲自去安装使用每一个软件，最终基于自己的经验以及需求达到一个最适合自己的选择。

原文：http://www.qyjohn.net/?p=1337

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

# 使用Eucalyptus打造自己的云测试平台

文/Rini Susan & Vikas Valikan
编译/黄永兵

本文介绍如何使用各种开源技术在云中搭建一个测试平台，你可以使用它作为一个指导建立你自己的云测试平台。下面是我搭建测试平台用到的开源技术：

安装 CentOS 5.2 的机器：它们将作为云，集群和节点控制器。

Eucalyptus 1.5.1

Apache Tomcat 6.0.14

Jakarta JMeter 2.3.2：这个开源工具的目的是执行负载测试和功能行为分析，以及测量应用程序性能，主要是 Web 应用程序。

MySQL 5.0

预封装的 CentOS 5.2 镜像

JPetStore Web 应用程序：这个简单的应用程序是 J2EE 平台在现实应用程序设计中的一个工作示范。

在动手之前，你需要了解云计算和 Eucalyptus 工作原理的基本知识。

## 云测试环境组件

我们的实验环境由四台机器组成，一台是 2GB 内存的机器，其它三台内存均是 1GB，全部安装 CentOS 5.2：

Eucalyptus 云和集群相关的 RPM 包安装在 2GB 内存的机器上，担任云和集群控制器；

其余三台机器作为节点控制器，只安装节点 RPM 软件包；

其中一台机器安装 JMeter 作为 JMeter 主服务器。

下面的镜像是在搭建测试环境时要使用到的：

Tomcat 镜像：用于应用程序部署。

MySQL 镜像：用于数据库部署。

JMeter 镜像：用于测试和监控。

## 为云测试环境创建镜像

这一部分我们将介绍如何创建前面列出的三种镜像，包含必要的步骤和相关脚本。你可以从任意 CentOS 5.2 机器创建镜像，在开始之前，我们需要一个预封装的 CentOS 5.2 镜像。

## 创建Tomcat镜像

预封装的 CentOS 镜像已经挂载到本地目录：

```
mkdir /mnt/Mount
```

接下来将预封装的 CentOS 5.2 镜像挂载到创建的目录上。

```
mount -o loop /mnt/Mount
mount -t proc none /mnt/Mount/proc/
```

将 Tomcat 安装到挂载目录中(例如，将 Tomcat 文件夹放入 /mnt/

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

Mount/home)。为了确保 Tomcat 随系统自动启动,需要编辑 rc.local 文件。

我们用于测试的 Web 应用程序是 JPetStore,将 jpetstore.war 放入 /webapps/ 文件夹,现在镜像包含所有必要的软件和脚本了。

接下来卸载镜像：

umount /mnt/Mount/proc

umount -d /mnt/Mount

现在得到的镜像包含 Tomcat 6 和 Web 应用程序,我们可以将它上传到云中了。

### 创建JMeter镜像

创建 JMeter 镜像的步骤和前面创建 Tomcat 镜像的步骤一样。

挂载镜像,将 JMeter 2.3.2 文件移动到 /mnt/Mount/home 文件夹下。JMeter 主服务器安装在一台物理 Linux 机器上, JMeter 实例作为从服务器。首先,将主服务器添加到从服务器的“已知主机”列表中;然后为主服务器生成一个无密码密钥,并将其添加到从镜像,将主服务器上产生的 id_dsa 放在从镜像的指定文件夹中(如 /mnt/Mount/home)。每当从实例启动时,在已知主机列表中就会有主服务器的 IP。

当 JMeter 从实例启动时,主服务器的 jmeter.properties 文件需要更新它的 IP 地址,为了自动添加 IP 地址,我们使用了一个 Shell 脚本,在挂载文件夹中创建一个脚本文件(/mnt/Mount/home)：

(脚本内容略)

将脚本保存为 .sh 文件,在 rc.local 文件中添加实例启动时需要自动启动的服务路径。

接下来卸载掉镜像。

### 创建MySQL镜像

创建 MySQL 镜像的步骤和创建 Tomcat 镜像的步骤一样。

挂载镜像,修改根挂载点(如 /mnt/Mount),然后执行下面的命令：

cd /mnt/Mount

chroot .

接下来安装 MySQL 需要的 RPM 包,包括依赖包。

安装完所有 RPM 包后,从 root 退出,启动 mysqld。为了让 MySQL 也随系统自动启动,在 /etc/rc.local 文件中添加启动脚本。

接下来配置 JPetStore 使用新的 MySQL 实例：

```
driver=org.gjt.mm.mysql.Driver
url=jdbc:mysql://:3306/JPETSTORE
username=
password=
```

当实例启动和运行时,你可以按照我们的要求,将 JPetStore 的默认数据库修改为 MySQL,在 Tomcat 镜像中,更新 JPetStore 的 database.properties 文件。

启动所有实例,确保每个实例都获得了一个 IP 地址。启动好后,你就可以在这些云实例上执行性能测试,测试方法和在物理机上执行性能测试没有区别。

由于篇幅所限,本文为删节版,完整内容见原文：

Building a Test Platform in the Cloud with Open Source Technologies

译文 :http://os.51cto.com/art/201010/231601.htm

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

# 以公司实际应用讲解OpenStack到底是什么

文/livemoon

OpenStack 是一个云平台管理的项目，它不是一个软件。这个项目由几个主要的组件组合起来完成一些具体的工作。要想直观的了解它是什么样子的，请参阅《OpenStack 详细解读 :定义，好处与使用实例》一文。

就目前而言，OpenStack 在国外慢慢的流行开来，不少企业和个人也在对它进行二次开发。从我个人理解，OpenStack 作为一种免费的开源软件，可以用在中小企业内部，可以给公司内部的开发测试部门使用，也可以跑一些应用服务。另外一种就是提供对外服务，好比作云服务的企业会考虑对 OpenStack 进行二次开发和包装，集成或者新增一些特定的功能或者管理界面。我觉得 OpenStack 不光光能在 1 分钟给你想要的 image 操作系统，也可以做到 5 分钟能帮你生成一台 app 节点（应用服务器）加入到业务中去。后者才是我们现在更需要去做的，从 irc 聊天室、邮件列表、以及一些 wiki 的内容来看，老外已经在这方面走在了前面。

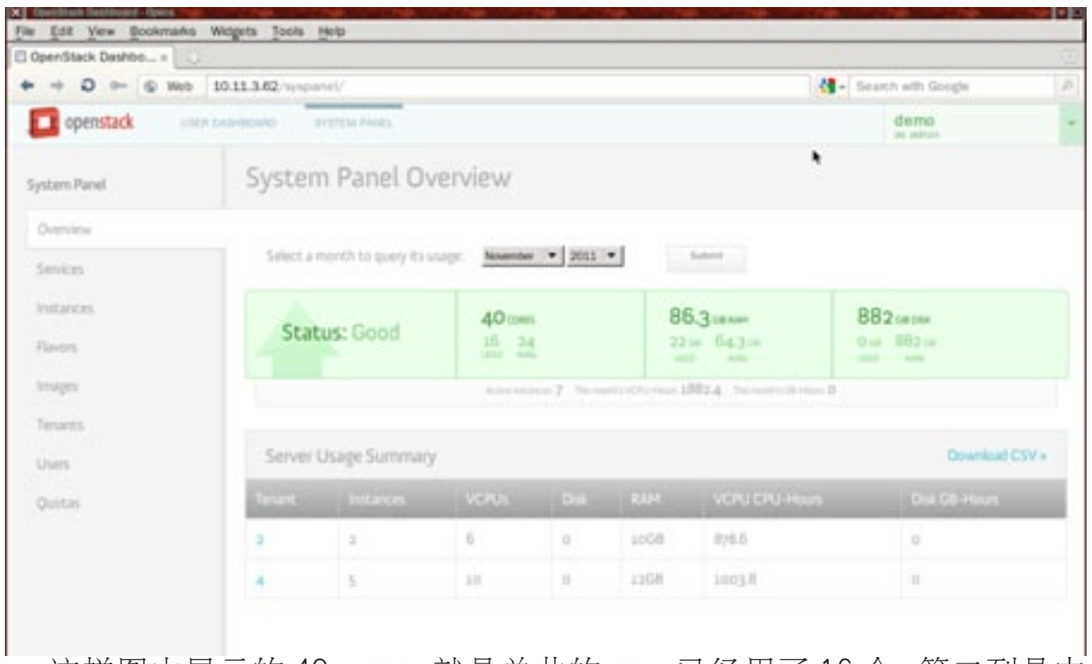
在这篇文章里，我将介绍一下一个简单的、可用在公司内部的 OpenStack 构建起来的管理平台。它看上去如右图：

这个环境一共用了 6 台 8 核的服务器。除去控制器的核心不算，一共有 40 个可用于计算的核心。其中：

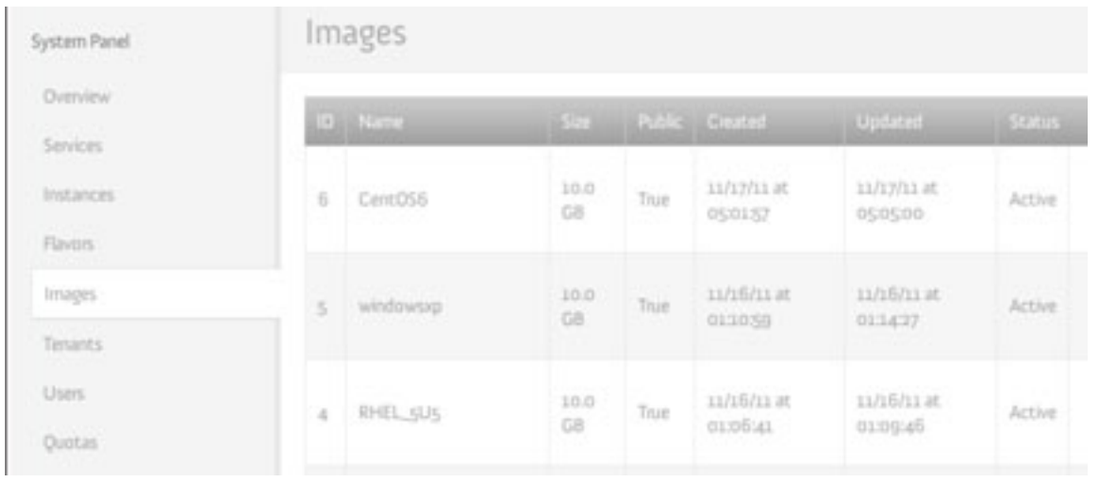
启动了 4 台的 cpu 作为计算节点用来跑虚拟机（nova-compute）

一台服务器安装了 nova，glance，keystone，dashboard 的所有服务和 mysql 数据库作为控制节点

一台启用了 nova-volume 服务，提供给虚拟机额外的块存储



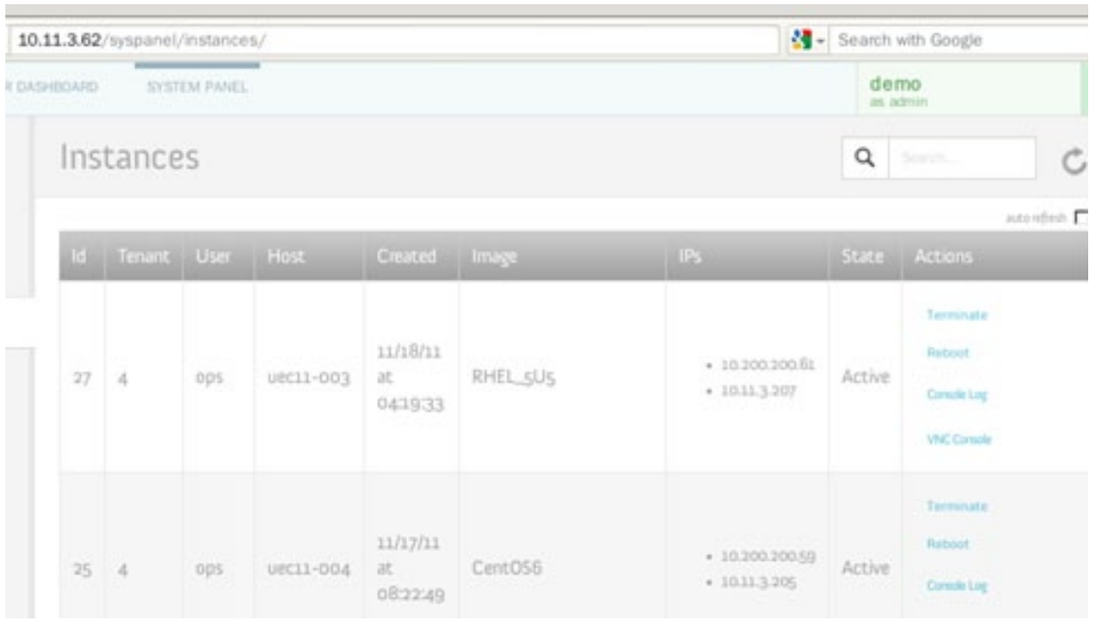

这样图中显示的 40 cores 就是总共的 cpu，已经用了 16 个；第二列是内存，下面显示了有两个部门。分别跑了 2 个和 5 个实例。



杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

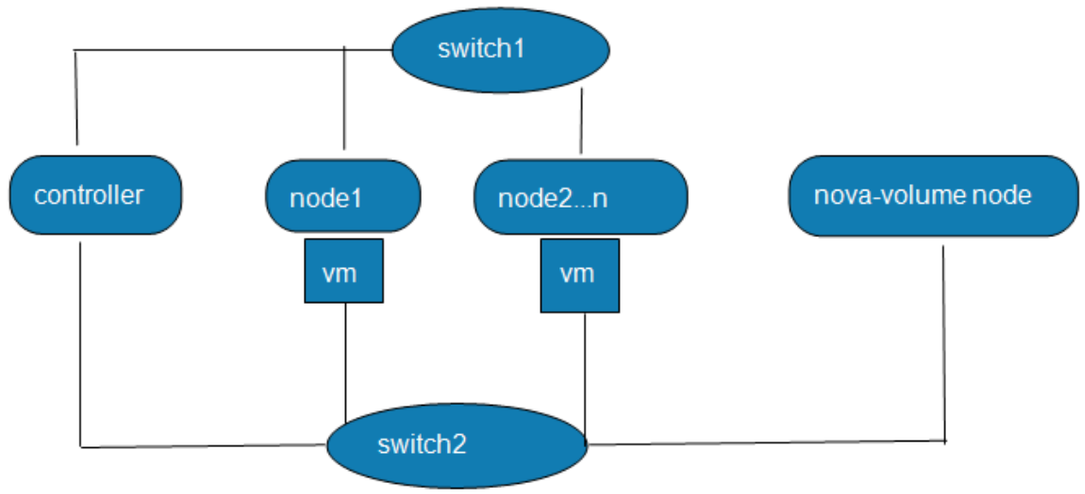
上图展示的是 Images，通俗的讲就是预先做好的系统或者模板。images 是通过名叫 glance 的这个组件来管理（这下知道 glance 的用处了吧），它提供命令接口允许用户把自己做好的系统（支持 img，qcow2 等格式），至于如何用 kvm 做自己的 img，可以参考这份文档。

在图中可以看出，有 CentOS，Windows XP，RHEL 的模板。另外 3 个 image 是用户自己做的，简单的是就是用户使用我做的 RHEL（里面只装了一些基本的系统软件）生成虚拟机实例，然后在虚拟机中配置了他自己需要的软件应用。配置完成之后，保存为 rhel_app 这个 image，这样下次有需要的时候，就可以直接从 rhel_app 启动新的实例，1 分钟之内就可以使用他需要的应用。




上图显示的就是目前跑在私有云上的实例。我们可以看到右边有四个选项，Terminate 是撤销，也就是删除虚拟机实例，Reboot 重启，Console Log 显示终端上的信息，VNC Console 这个是直接在 web 上面开个 vnc 窗口显示 console，另外还有 Snapshot 的按钮，这个按钮会出现在以用户自己身份登陆的界面上。目前我是以 admin 身份登陆。

限于篇幅原因，还有很多 tab 页面我不做介绍了。总的来说，你只要给一个用户一个帐号，他就能从 image 选择不同配置（cpu，内存，磁盘）的实例，分配 ip，开端口，登陆，完全自主的操作，不需要管理员去干涉。如果你觉得这套管理工具对你或者你们企业来说有一定的帮助，想要尝试一下，或者基于它来作二次开发（因为 OpenStack 是完全开源的），可以继续往下看，我将会简单介绍一下如何构造这么一个系统。




上图是个简单的拓扑图。每台 host 都有两块网卡，连接 switch1 的是外部访问接口，就是用户可以直接连接到的 ip 网络，这个网络用来提供给虚拟机以便用户使用。switch2 使用一个内部的网络，即对用户不可见，我们可以设定一个私有网络，这个网络用来 node 节点和 controller 之间的网络通讯，image 的传输，nova—volume 和 node 之间的 iscsi 的数据传输。

环境准备

所有的服务器都安装 Ubuntu 11.10。

由于篇幅所限，本文删节了构造步骤的部分，完整内容见原文：

http://os.51cto.com/art/201111/303120.htm

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

# 为 OpenNebula 制作 Ubuntu 镜像

文/vpsee

为 OpenNebula 制作 Ubuntu 镜像的步骤和制作 Windows 镜像差不多，以下是具体步骤：

首先下载需要安装的 ubuntu 版本：

```
$ wget http://releases.ubuntu.com/11.10/ubuntu-11.10-server-amd64.iso
```

创建一个 10GB 大小的“硬盘”（raw 格式）：

```
$ kvm-img create -f raw ubuntu.img 10G
Formatting 'ubuntu.img', fmt=raw size=10737418240
```

然后使用刚才下载的 ubuntu “安装盘”和刚创建的“硬盘”引导启动系统，使用 –vnc 参数打开 vnc 访问，以便可以用其他机器远程登录进行安装操作：

```
$ sudo kvm -m 512 -cdrom ubuntu-11.10-server-amd64.iso \
-drive file=ubuntu.img -boot d -nographic -vnc :0
```

在其他机器上用 vnc 客户端登录后就可以看到 Ubuntu 安装界面，按照屏幕的提示完成 ubuntu 的安装工作，需要注意的是分区的时候只分一个区给 /，不要分 swap 区，以后 VPSee 将会提到如何给虚拟机加上交换分区：

```
$ vncviewer 172.16.39.111:5900
```

安装完后会自动重启，shutdown –h now 虚拟机后再按照下面命令启动刚刚安装好的虚拟机镜像 ubuntu.img，如果出现 failed to find romfile “pxe-rtf8139.bin” 的错误提示可以通过安装 kvm-pxe 解决：

```
$ sudo kvm -m 512 -drive file=ubuntu.img -boot c -nographic -vnc :0
kvm: pci_add_option_rom: failed to find romfile "pxe-rtl8139.bin"
$ sudo apt-get install kvm-pxe
```

再次用 vnc 登录虚拟机镜像，升级和更新系统，可以安装一些必要工具，比如 OpenSSH 之类的：

```
$ vncviewer 172.16.39.111:5900
$ sudo update
$ sudo upgrade
$ sudo apt-get install openssh-server
```

创建和编辑虚拟网络配置文件，然后创建一个 OpenNebula 虚拟网络（参考：在 CentOS 上安装和配置 OpenNebula）：

```
$ vi small_network.net
NAME = "Small network"
TYPE = FIXED

BRIDGE = br0
LEASES = [ IP="172.16.39.111"]
```

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

```
LEASES = [ IP="172.16.39.112"]
LEASES = [ IP="172.16.39.113"]
```

```
$ onevnet create small_network.net
```

```
$ onevnet list
  ID USER     NAME            TYPE BRIDGE P #LEASES
   0 oneadmin Small network    Fixed    br0 N       0
```

创建和编辑 Ubuntu 虚拟机的启动配置文件。注意别忘了加上 ARCH = x86_64（否则无法正常启动 Ubuntu），我们刚才安装的是 Ubuntu 64 位 Server 版（ubuntu-11.10-server-amd64.iso）：

```
NAME   = ubuntu
CPU    = 1
MEMORY = 512

OS = [ ARCH = x86_64,
       BOOT = hd,
       ROOT = sda1
]

DISK = [ source   = /var/lib/one/images/ubuntu.img,
         clone    = no,
         target   = sda,
         readonly = no ]

GRAPHICS = [ type ="vnc",
             listen ="0.0.0.0",
             port = "5900" ]

NIC    = [ NETWORK = "Small network" ]
```

依照上面的配置在 OpenNebula 上创建一个 Ubuntu 虚拟机，等待一下 OpenNebula 会自动根据当前资源情况调度，期间不断用 onevm list 命令查看当前虚拟机的创建情况，状态会从 pend -> prol -> boot -> runn，runn 状态就表示虚拟机已经成功创建并正常运行。最后检查一下 OpenNebula 是否成功创建一个名叫 ubuntu 的虚拟机：

```
$ onevm create ubuntu.one
```

```
$ onevm list
  ID     USER     NAME STAT CPU     MEM        HOSTNAME        TIME
  42 oneadmin    ubuntu runn   1     512M         node00 00 01:16:39
```

原文：http://www.vpsee.com/2011/12/create-ubuntu-kvm-image-for-opennebula/

相关阅读：

linux 光盘镜像文件制作攻略

如何创建基本的虚拟镜像

MED-V 动手实验四：虚拟机镜像制作

在 CentOS 上安装和配置 OpenNebula

Ubuntu 11.04 Server 安装配置 OpenNebula 3.0

安装 OpenNebula 基于 Web 的管理控制台

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

# 使用CONVIRT管理基于KVM的虚拟机——安装篇

文/hetao

ConVirt 是一个直观的、图形化的虚拟机管理工具，可以对虚拟机的整个生命周期进行管理。

KVM 是使用于 Linux 内核中的虚拟化基础设施。KVM 目前支持 Intel VT 及 AMD-V 的原生虚拟技术。KVM 在 2007 年 2 月被导入 Linux 2.6.20 内核中。它也被引入 FreeBSD。在 Mac OS X 中，也可以看见 KVM。

**资源：**

三台计算机（CPU：Intel VT 或 AMD-V），一台作为管理终端，一台作为被管理的虚拟资源服务器（虚拟机安装在机上），另外一台用来安装 ConVirt。

操作系统安装盘，Ubuntu Server 10.04.2 (Lucid Lynx) x86_64 和 Ubuntu Desktop 10.04.2 (Lucid Lynx) x86_64。

**安装步骤：**

为三台计算机安装操作系统

用于管理的计算机安装 Ubuntu Desktop 10.04.2 (Lucid Lynx) x86_64，而用于安装 ConVirt 和用于安装虚拟机的计算机则安装 Ubuntu Server 10.04.2 (Lucid Lynx) x86_64。

配置第三方源

编辑软件包源列表文件 /etc/apt/sources.list，追加 http://archive.canonical.com/ubuntu lucid partner，并更新软件索引：

```
sudo apt-get update
```

安装被管理的虚拟资源服务器（使用两台 Ubuntu Server 中的一台）

安装 KVM

```
sudo apt-get install ssh kvm socat dnsmasq uml-utilities lvm2 expect
```

配置虚拟资源服务器：

通过安装 convirture-tools 来帮助你配置虚拟资源服务器，使得其可以通过 ConVirt 来进行方便的管理。该命令将创建一个公有的网桥，相关的脚本并将操作摘要写入 /var/cache/convirt/server_info。

```
sudo apt-get install convirture-tools
```

安装相关依赖：

```
sudo convirt-tool install_dependencies
```

配置网络：

```
sudo convirt-tool setup
```

安装和配置 ConVirt

安装：

```
sudo apt-get install convirt2
```

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

配置防火墙，使得可以通过 VNC 来连接虚拟机控制台。

iptables -I INPUT -m state --state NEW -p tcp --dport 6900:6999 -j ACCEPT

配置 VNC：

添加 SSH Key，使得从 ConVirt 到被管理的虚拟资源服务器的 SSH 连接采用 Key 的方式进行认证。

```
cp /var/lib/convirt/identity/cms_id_rsa.pub /root/.ssh/id_rsa.pub
cp /var/lib/convirt/identity/cms_id_rsa /root/.ssh/id_rsa
scp /var/lib/convirt/identity/cms_id_rsa.pub root@managed-server/root/.ssh/
  cms_id_rsa.pub
ssh root@managed-server
cat ~/.ssh/cms_id_rsa.pub >> ~/.ssh/authorized_keys
```

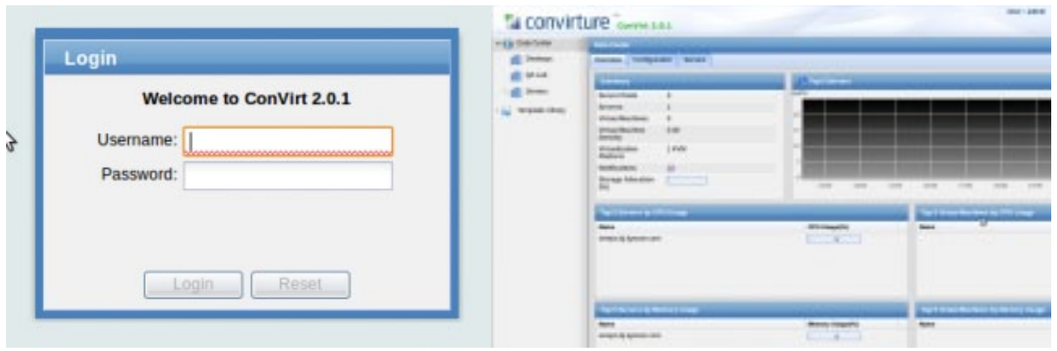
## 登录ConVirt管理系统

启动 Convirt：

sudo convirt-ctl start

登录 Convirt（用户名 / 密码 admin/admin）：

http://localhost:8081




原文：via hetao.im

【特别推荐】

# OpenStack Nova 完整安装手册

LinuxTone 的 yz 在 11 月发布了一份非常详尽的 OpenStack Nova 安装手册，内容新鲜，绝对值得收藏！

目录（简版）：

◆实验环境

◆架构部署

◆服务器系统安装

◆控制节点安装

◆计算节点安装

◆ DASHBOARD 使用基础

LinuxTone 下载地址：

http://bbs.linuxtone.org/thread-17390-1-1.html

51CTO 下载地址：

http://down.51cto.com/data/306764

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

# MySQL主从配置的一些总结

文/抚琴煮酒

一、做了 MySQL 主从也有一段时间了，这两天检查磁盘空间情况，发现放数据库的分区磁盘激增了 40 多 G，一路查看下来，发现配置好主从复制以来到现在的 binlog 就有 40 多 G，原来根源出在这里，查看了一下 my.cnf，看到 binlog 的 size 是 1G 就做分割，但没有看到删除的配置，在 MySQL 里 show 了一下 variables：

mysql>show variables like '%log%';

查到了，

| expire_logs_days | 0 |

这个默认是 0，也就是 logs 不过期，这个是一个 global 的参数，所以需要执行

set global expire_logs_days=8;

这样 8 天前的 log 就会被删除了，如果有回复的需要，请做好备份工作，但这样设置还不行，下次重启 mysql 了，配置又回复默认了，所以需在 my.cnf 中设置，

expire_logs_days = 8

这样重启也不怕了。

现在我在生产环境下的做法是将此时间设为 0，然后备份 mysql 日志文件，然后再手动清理此文件。

想要恢复数据库以前的资料，执行

mysql>show binlog events;

由于数据量很多，查看起来很麻烦，光打开个文件就要闪半天，所以应该适当删除部分可不用的日志。

并且如果使用的时间足够长的话，会把我的硬盘空间都给吃掉。

1、登录系统，/usr/bin/mysql

使用 mysql 查看日志：

```
mysql>show binary logs;
+————————-+———— -+
| Log_name | File_size |
+————————-+———— -+
| ablelee.000001 | 150462942 |
| ablelee.000002 | 120332942 |
| ablelee.000003 | 141462942 |
+————————-+———— -+
```

2、删除 bin-log（删除 ablelee.000003 之前的而没有包含

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

ablelee.000003）：

```
mysql> purge binary logs to ´ ablelee.000003 ´ ;
Query OK, 0 rows affected (0.16 sec)
```

3、查询结果（现在只有一条记录了）：

```
mysql> show binlog events\G
*************************** 1. row ***************************
Log_name: ablelee.000003
Pos: 4
Event_type: Format_desc
Server_id: 1
End_log_pos: 106
Info: Server ver: 5.1.26-rc-log, Binlog ver: 4
1 row in set (0.01 sec)
(ablelee.000001 和 ablelee.000002 已被删除 )
mysql> show binary logs;
+————————-+————— – +
| Log_name | File_size |
+————————-+————— – +
| ablelee.000003 | 106 |
+————————-+————— – +
1 row in set (0.00 sec)
( 删除的其它格式运用 !)
PURGE {MASTER | BINARY} LOGS TO ‘log_name’
PURGE {MASTER | BINARY} LOGS BEFORE ‘date’
```

用于删除在指定的日志或日期之前的日志索引中的所有二进制日志。这些日志也会从记录在日志索引文件中的清单中被删除，这样被给定的日志成为第一个。

例如：

```
PURGE MASTER LOGS TO 'mysql-bin.010';
PURGE MASTER LOGS BEFORE '2008-06-22 13:00:00';
```

二、现在手上蛮多项目的数据库用的是 MySQL，由于权限等原因，暂时不方便部署 Nagios 监控 MySQL 主从复制，所以我一般在从机上配置了 SHELL 脚本用来监控 MySQL 的主从状态（设置为每十分钟运行一次），并且每次出问题时将确切日期写进错误日志，方便事后排查原因，脚本内容如下：

(脚本内容略)

脚本设计思路：

1、此脚本应该能适应各种各样不同的内外网环境，即 IP 不同的环境；

2、让脚本也顺便监控下 MySQL 是否正常运行；

三、innodb_buffer_pool_size 的设置。

这个参数定义了 InnodDB 存储引擎的表数据和索引数据的最大内存缓冲区大小。和 MyISAM 存储引擎不同，MyISAM 的 key_buffer_size 只缓存索引键，而 innodb_buffer_pool_size 却是同时为数据块和索引块 做缓存，这个特征和 Oracle 是一样的，这个值设得越高，访问表中数据需求的 I/O 就越少。在一个专用的数据库服务器，可以设置这个参数达机器物理内存的 80%，我现在一般的做法是配置成物理内存的 1/4，比如 8G 内存的生产数据库，我一般会配置成 2G 左右。

由于篇幅所限，本文有删节，完整内容见原文：

http://database.51cto.com/art/201111/304473.htm

推荐作者图书：《构建高可用 Linux 服务器》

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

# 智能DNS(Bind dlz)在企业中的应用

文/崔晓辉

去年因为二级域名大量增加、Bind 下管理不便的关系，在公司部署了智能 DNS（Bind dlz），当时写过一篇部署文档进行了记录。不过在过去一年的使用当中，发现了其中不少的错误，所以本次发布该文档的第二版，对这些错误进行修正。

## 一、Bind-dlz简介

全世界范围内标准 DNS 服务器是 BIND。尽管被流传了许多年，经过多次修改，BIND 的基本功能保持不变。遗憾的是，有一些不好的缺陷。

BIND 从文本文件中获取数据，这样容易因为编辑错误出现问题。

BIND 需要将数据加载到内存中，如果域或者记录较多，会消耗大量的内存。

BIND 启动时解析 Zone 文件，对于一个记录较多的 DNS 来说，会耽误更多的时间。

如果最近修改一条记录，那么要重新加载或者重启 BIND 才能生效，可能会影响客户端查询。

bind-dlz 主要解决上述缺陷而诞生，在 mysql 存储 zone 的记录，比在文本中好管理的多。

智能 DNS 的原理：

在用户解析一个域名的时候，判断一下用户的 IP，然后跟 DNS 服务器内部的 IP 表匹配一下，看看用户是电信还是网通用户，然后给用户返回对应的 IP 地址。

适用范围：

网站要有三线路接入或者在电信、联通、移动部署有服务器，这样智能 dns 才能派上用场。

## 二、智能DNS系统服务规划

1、NameServer 服务器设置 (到新网或者万网后台添加)

ns1.zjyxh.com 192.19.13.15

ns2.zjyxh.com 192.19.11.3

NS1 是 master ，NS2 是 slave。两者数据通过 mysql 来同步。

2、测试 NS 记录是否生效

```
#dig ns www.zjyxh.com
#dig www.zjyxh.com +trace
```

3、Bind-View 规划

www.zjyxh.com 网通（CNC） 124.133.11.78

www.zjyxh.com 电信（TELECOM） 58.56.11.153

www.zjyxh.com 移动（ANY） 120.192.11.13

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

## 三、在CentOS 5.7上安装MySQL Replication

因为 Bind-dlz 是使用 MySQL 作为存储 zone 的载体，这样就可以用 php 来操作 MySQL。特别注意：智能 dns 最少部署两台 NameServer，主从关系。主从同步利用 mysql 的复制来实现主从同步。

首先下载 mysql 的最新版并解压。将 my.cnf 放到 /etc 下，并安装系统数据库。

MySQL replication 配置

1、MySQL 安全设置

2、删除默认的数据库和用户。

我们的数据库是在本地，并且也只需要本地的 php 脚本对 mysql 进行读取，所以很多用户都不需要。mysql 初始化后会自动生成空用户和 test 库，这会对数据库构成威胁，我们全部删除。

3、Master 机器设置权限，赋予 Slave 机器 FILE 及 Replication Slave 权利，并打包要同步的数据库结构。

4、修改 Slave 服务器的 my.cnf

5、删除 Slave 端数据库目录中的 master.info

6、重启动 Slave 的 MySQL 服务。

7、测试

先检测两个 MySQL 数据库中的 cdn 是否正常。正常情况应该是 Master 和 Slave 中的 MySQL 都有相同的 cdn 数据库，并且里面的数据都一样。然后测试 replication 功能是否启用。

接下来是一些性能方面的调优。

为 MySQL 添加 TCMalloc 库降低系统负载

TCMalloc（Thread-CachingMalloc）是 google 开发的开源工具——“google-perftools”中的成员。与标准的 glibc 库的 malloc 相比，TCMalloc 在内存的分配上效率和速度要高得多，可以在很大程度上提高 MySQL 服务器在高并发情况下的性能，降低系统负载。

1、64 位操作系统请先安装 libunwind 库，32 位操作系统不要安装。libunwind 库为基于 64 位 CPU 和操作系统的程序提供了基本的堆栈辗转开解功能，其中包括用于输出堆栈跟踪的 API、用于以编程方式辗转开解堆栈的 API 以及支持 C++ 异常处理机制的 API。

2、安装 google-perftools：

3、修改 MySQL 启动脚本（根据你的 MySQL 安装位置而定）：

4、使用 lsof 命令查看 tcmalloc 是否起效：

## 四、安装配置Bind-DLZ 及相关脚本

1、安装 bind

2、创建相关配置文件

3、配置 DNSTSIG

## 注意事项

部署 DNS，防火墙和路由器要设置清楚，我部署的时候就是因为硬防没有对 master 和 slave 服务器开放 tcp 和 udp53 端口，造成不能解析域名。需要大家切记！

由于篇幅关系，本文仅列出了整体的实现步骤框架。具体实现、脚本与管理界面下载见原文：http://os.51cto.com/art/201111/305114.htm

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

# 针对Linux集群的高级监控工具sinfo概述

文/John Knight
编译/布加迪

你是不是面临这种情况：想搭建某种网络集群，但又要处理许多不同的计算机，想密切跟踪这所有计算机几乎是不可能的事？如果你负责满满一屋子的计算机，还要负责使用这些机器的那些用户，又该如何是好？ sinfo 也许正是你苦苦寻觅的那款工具。Freshmeat 网站上的介绍如下：

Sinfo 是一款监视工具，使用广播方案来发布关于你本地网络上每一台计算机的运行状况的信息。它支持显示多方面的内容，比如处理器、内存使用情况、网络负载以及关于每一台计算机上五个主要进程的信息。Sinfo 使用 ncurses，以一目了然的方式来显示信息。

Sinfo 可以显示关于多台计算机的系统信息，以便管理。使用的时候可以通过 –s 选项查看更多信息。

## 安装过程

如果你使用基于 Debian 的系统，比如 Debian 和 Ubuntu 等系统，可以使用二进制包，可以在你的 repo 中找一下。考虑到该软件包括了一个启动守护程序 sinfod，我强烈建议使用这个可选的二进制文件，因为这个过程的许多方面实现了自动化（它也是我在这里探讨的版本）。不过，为了确保发行版中立，与往常一样，我还在安装过程中介绍了源版本。

说明文档对代码库的要求如下：

ncurses：用于终端处理的代码库（5.7 版本）。

boost：可移植的 C++ 源代码库，使用 Boost.Bind 和 Boost.Signals（1.42 版本）。

asio（>=1.1.0）：asio 是一个跨平台的 C++ 代码库，用于网络编程（1.4.1 版本）。

如果你通过源代码编译，还需要上面这些代码库的开发包（–dev）。libboost– 下的开发包的数量相当多，所以要是你在编译过程中遇到了任何问题，请先检查 libboost 是不是安装全了。

对于使用源代码来运行的那些人来说，一旦你搞定了代码库要求，就可以获取最新的 tarball 文件（下载地址）。解压缩，在新的文件夹中打开终端，输入以下命令：

```
$ ./configure

$ make
```

如果你的发行版使用 sudo：

```
$ sudo make install
```

如果你的发行版使用 root：

```
$ su

# make install
```

在我继续下文之前，应该解释一下：sinfo 分平常的应用软件部分和后台守护程序这两个部分。守护程序的安装每个发行版都不同，这部分我就不

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

具体说了，细节可以查看源代码 tarball 文件的使用说明文件和官方网站。

## 使用

Sinfo 是一款“半图形用户界面 (GUI)”的命令行程序，使用起来实际上很容易，不过高级用户会通过命令行的参数选项符让它处理一些相当出色的任务。想让该程序在基本模式下运行，只要输入：

```
$ sinfo
```

如果你只是在自己的机器上安装了 sinfo，显示的信息将仅是你这台机器的信息。你可以从这个屏幕上看到可用内存、处理器占用率和主机名称等信息。文末的附录列出了适用的快捷键命令，只要按一个键，就可以切换该程序的不同部分。

不过在这种情况下，sinfo 其实只是更漂亮的 top 而已。使用 sinfo 的目的是，你可以一下子显示来自好几台机器的信息，以便密切监视局域网的运行情况。

要做到这一点也很容易：只要在网络上的其他计算机上也安装并运行 sinfo，运行之后你就会发现两台计算机的信息分别在两个计算机上都有显示。继续把它安装到其他联网计算机上，显示列表就会越来越长。

这些只是基本功能，更丰富的功能方面又如何呢？很显然，因篇幅所限，我没法在这里一一介绍 (你其实应该查阅参考手册页，了解更多详细内容)，不妨看一下我偏爱使用的一些功能。

在命令行，如果你添加了 -W 参数选项符 (或者 --wwwmode)，就像这样：

```
$ sinfo -W
```

输出就会从平常的类似 top 的屏幕变出 HTML 输出——对于喜欢借助自动化网页等方面进行远程管理的那些人来说，这非常方便。

在编写某种命令行脚本时，你可以添加参数选项符 -s (或者 --ysteminfo) 输出一大段重要的系统信息。举例来说，我的两台机器显示了以下的额外信息：

```
192.168.1.2   knightro-bigdesktop i686
    Linux 2.6.32-27-generic #49-Ubuntu SMP Wed De
cpus: 4  MHz:  800.0
RAM: 3276.5 MByte   swap: 7629.4 Mbyte
load 1min:  0.0   load 5min:  0.1   load 15min:  0.1

192.168.1.1   nhoj-desktop x86_64
    Linux 2.6.38-8-generic #42-Ubuntu SMP Mon Apr 11 0
cpus: 2  MHz: 1000.0
RAM: 2007.6 MByte   swap: 2047.3 Mbyte
load 1min:  0.1   load 5min:  0.2   load 15min:  0.1
uptime    0 days, 19:13:03
```

这样一种信息表明 sinfo 有许多潜在用途，我立即想到了可以在局域网派对 (LAN party) 上用于监视和故障排除。要是任何一个节点有问题，主机在试图隔离这个问题时很可能就能够立即着手处理。

## 结语

sinfo 设计精巧，安装方便，我认为这款程序会很快闯出自己的一片天地。但愿它会像其他标准应用软件那样变得司空见惯，成为一款常用工具。也许对它进行移植就能实现这个目标。

原文：sinfo–Advanced Network Monitoring

译文：http://os.51cto.com/art/201112/305816.htm

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

# 系统管理员必须知道的PHP安全实践

文/ VIVEK GITE
编译/布加迪

Apache web 服务器提供了这种便利：通过 HTTP 或 HTTPS 协议，访问文件和内容。配置不当的服务器端脚本语言会带来各种各样的问题。所以，使用 PHP 时要小心。以下是 25 个 PHP 安全方面的最佳实践，可供系统管理员们安全地配置 PHP。

为 PHP 安全提示而提供的示例环境

◆文件根目录 (DocumentRoot)：/var/www/html

◆默认的 Web 服务器：Apache（可以使用 Lighttpd 或 Nginx 来取代 Apache）

◆默认的 PHP 配置文件：/etc/php.ini

◆默认的 PHP 加载模块配置目录：/etc/php.d/

◆我们的示例 php 安全配置文件：/etc/php.d/security.ini（需要使用文本编辑器来创建该文件）

◆操作系统：RHEL/CentOS/Fedora Linux（相关指令应该与 Debian/Ubuntu 等其他任何 Linux 发行版或者 OpenBSD/FreeBSD/HP-UX 等其他类似 Unix 的操作系统兼容）。

◆默认的 php 服务器 TCP/UDP 端口：无

## 将所有PHP错误记入日志

别让 PHP 错误信息暴露在网站的所有访客面前。编辑 /etc/php.d/security.ini，执行以下指令：

```
display_errors=Off
```

确保你将所有 PHP 错误记入到日志文件中：

```
log_errors=On
error_log=/var/log/httpd/php_scripts_error.log
```

## 不允许上传文件

出于安全原因，编辑 /etc/php.d/security.ini，执行以下命令：

```
file_uploads=Off
```

如果使用你应用程序的用户需要上传文件，只要设置 upload_max_filesize，即可启用该功能，该设置限制了 PHP 允许通过上传的文件的最大值：

```
file_uploads=On
# 用户通过 PHP 上传的文件最大 1MB
upload_max_filesize=1M
```

## 关闭远程代码执行

如果启用，allow_url_fopen 允许 PHP 的文件函数——如 file_get_contents()、include 语句和 require 语句——可以从远程地方 (如 FTP 或网站)

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

获取数据。

allow_url_fopen 选项允许 PHP 的文件函数——如 file_get_contents()、include 语句和 require 语句——可以使用 FTP 或 HTTP 协议，从远程地方获取数据。编程员们常常忘了这一点，将用户提供的数据传送给这些函数时，没有进行适当的输入过滤，因而给代码注入安全漏洞留下了隐患。基于 PHP 的 Web 应用程序中存在的众多代码注入安全漏洞是由启用 allow_url_fopen 和糟糕的输入过滤共同引起的。编辑 /etc/php.d/security.ini，执行以下指令：

```
allow_url_fopen=Off
```

出于安全原因，我还建议禁用 allow_url_include：

```
allow_url_include=Off
```

## 启用SQL安全模式

编辑 /etc/php.d/security.ini，执行以下指令：

```
sql.safe_mode=On
```

如果启用，mysql_connect() 和 mysql_pconnect() 就忽视传送给它们的任何变量。请注意：你可能得对自己的代码作一些更改。sql.safe_mode 启用后，第三方开源应用程序（如 WorkdPress）及其他应用程序可能根本运行不了。我还建议你针对所有安装的 php 5.3.x 关闭 magic_quotes_gpc，因为它的过滤并不有效、不是很可靠。mysql_escape_string() 和自定义过滤函数能起到更好的作用：

```
magic_quotes_gpc=Off
```

## 控制POST请求的大小

作为请求的一部分，客户机（浏览器或用户）需要将数据发送到 Apache Web 服务器时，比如上传文件或提交填好的表单时，就要用到 HTTP POST 请求方法。攻击者可能会企图发送过大的 POST 请求，大量消耗你的系统资源。你可以限制 PHP 将处理的 POST 请求的最大大小。编辑 /etc/php.d/security.ini，执行以下命令：

```
; 在此设置实际可行的值
post_max_size=1K
```

1K 设置了 php 应用程序允许的 POST 请求数据的最大大小。该设置还影响文件上传。要上传大容量文件，这个值必须大于 upload_max_filesize。我还建议你限制使用 Apache Web 服务器的可用方法。编辑 httpd.conf，执行针对文件根目录 /var/www/html 的以下指令：

```
<Directory /var/www/html>
    <LimitExcept GET POST>
        Order allow,deny
    </LimitExcept>
## 可在此添加配置的其余部分 ... ##
</Directory>
```

## 资源控制（拒绝服务控制）

你可以设置每个 php 脚本的最长执行时间，以秒为单位。另一个建议的选项是设置每个脚本可能用于解析请求数据的最长时间，以及脚本可能耗用的最大内存数量。编辑 /etc/php.d/security.ini，执行以下命令：

```
# 设置，以秒为单位
max_execution_time = 30
max_input_time = 30
memory_limit = 40M
```

杂志订阅：http://os.51cto.com/art/201011/233915.htm
《Linux 运维趋势》投稿信箱：yangsai@51cto.com

## 为PHP安装Suhosin高级保护系统

Suhosin 是一款高级的保护系统，面向安装的 PHP。它旨在保护服务器和用户，远离 PHP 应用程序和 PHP 核心中的已知缺陷和未知缺陷。Suhosin 分两个独立部分，可以单独使用，也可以组合使用。第一个部分是针对 PHP 核心的小补丁，实施了几个低级防护措施，以防范缓冲器溢出或格式字符串安全漏洞；第二个部分是功能强大的 PHP 加载模块，实施了其他所有的保护措施。

## 保持PHP、软件和操作系统版本最新

打安全补丁是维护 Linux、Apache、PHP 和 MySQL 服务器的一个重要环节。应该使用以下其中任何一个工具（如果你通过软件包管理器来安装 PHP），尽快检查所有的 PHP 安全更新版本，并尽快打上：

```
# yum update 或
# apt-get update && apt-get upgrade
```

你可以配置红帽 /CentOS/Fedora Linux，以便通过电子邮件发送 yum 软件包更新通知。另一个选项是通过 cron job（计划任务）打上所有的安全更新版。在 Debian/Ubuntu Linux 下，可以使用 apticron 来发送安全通知。

注：经常访问 php.net，寻找源代码安装的最新版本。

## 限制文件和目录访问

确保你以 Apache 或 www 等非根用户的身份来运行 Apache。所有文件和目录都应该归非根用户（或 apache 用户）所有，放在 /var/www/html 下：

```
# chown -R apache:apache /var/www/html/
```

/var/www/html/ 是个子目录，这是其他用户可以修改的文件根目录，因为根目录从来不在那里执行任何文件，也不会在那里创建文件。

确保在 /var/www/html/ 下，文件权限设成了 0444（只读）：

```
# chmod -R 0444 /var/www/html/
```

确保在 /var/www/html/ 下，所有目录权限设成了 0445：

```
# find /var/www/html/ -type d -print0 | xargs -0 -I {} chmod 0445 {}
```

关于设置合适文件权限的补充

chown 和 chmod 命令确保：不管在什么情况下，文件根目录或文件根目录里面的文件都可以被 Web 服务器用户 apache 写入。请注意：你需要设置对你网站的开发模型最合理的权限，所以可以根据自身需要，随意调整 chown 和 chmod 命令。在这个示例中，Apache 服务器以 apache 用户的身份来运行。这可以在你的 httpd.conf 文件中用 User 和 Group 命令来配置。apache 用户需要对文件根目录下的所有内容享有读取访问权，但是不应该享有写入访问权。

确保 httpd.conf 有以下命令，实现限制性配置：

```
<Directory />
    Options None
    AllowOverride None
    Order allow,deny
</Directory>
```

由于篇幅所限，本文仅节选了 25 条最佳实践当中的 9 条。完整内容见原文：

25 PHP Security Best Practices For Sys Admins

译文：http://os.51cto.com/art/201111/305014.htm

# 招募启事

《Linux 运维趋势》的建设需要您的加入！

您可以通过如下方式参与我们杂志的建设：

1、推荐文章

无论是您在互联网上看到的好文章，还是您自己总结／整理的资料；无论是英文还是中文；无论是入门的还是高端的，都欢迎推荐！您可以直接在《Linux 运维趋势》新浪微群中分享：

http://q.weibo.com/121303

2、投稿

如果您愿意与大家分享您技术经验的热诚，那么欢迎您的投稿！原创或译文均可，稿件在 51CTO 首发可领取稿酬：）

投稿信箱：yangsai@51cto.com

3、推广与意见

如果您喜欢我们的杂志，认为这本杂志对于您的工作有所帮助，请向您的 Linux 好友、同事们推荐它！如果您觉得这份杂志还有什么地方需要改进或补充，也希望您能够提出您的宝贵意见！

反馈可至《Linux 运维趋势》新浪微群：

http://q.weibo.com/121303

或在新浪微博

@51CTO 系统频道


Linux 运维趋势

本刊发布日期：每个月的第二个星期五

您可以通过如下方式检查新刊发布：

1、电子邮件订阅：http://os.51cto.com/art/201011/233915.htm

2、RSS 订阅：http://www.51cto.com/php/rss.php?typeid=777

3、iPad 订阅：在《读览天下》客户端中可以搜索下载新刊到本地阅读！

本期杂志封面由 lazycai 制作


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