



# Netkiller Architect 手札

## Multi-dimension Architecture Design & Implementation

Mr. netkiller 陈景峰 (Neo Chan)

中国广东深圳宝安区龙华镇苹果园518049  
+86 755 29812080  
+86 755 29812080  
<[openunix@163.com](mailto:openunix@163.com)>

文档尚未完成，请勿转载！  
版权 © 2008, 2009, 2010, 2011 Copyright Editor Groups, All Rights Reserved

版权声明  
转载请与作者联系，转载时请务必标明文章原始出处和作者信息及本声明。



文档出处: <http://netkiller.sourceforge.net/> | <http://netkiller.github.com>  
文档最近一次更新于 Thu Dec 1 12:49:30 UTC 2011

下面是我多年积累下来的经验总结，整理成文档供大家参考:

- |  |   |  |   |
|--|---|--|---|
| <a href="#">Netkiller Architect 手札</a> | <a href="#">Netkiller Linux 手札</a>      | <a href="#">Netkiller Developer 手札</a> | <a href="#">Netkiller Database 手札</a>     |
| <a href="#">Netkiller Debian 手札</a>    | <a href="#">Netkiller CentOS 手札</a>     | <a href="#">Netkiller FreeBSD 手札</a>   | <a href="#">Netkiller Shell 手札</a>        |
| <a href="#">Netkiller Web 手札</a>       | <a href="#">Netkiller Monitoring 手札</a> | <a href="#">Netkiller Storage 手札</a>   | <a href="#">Netkiller Mail System 手札</a>  |
| <a href="#">Netkiller MySQL 手札</a>     | <a href="#">Netkiller LDAP 手札</a>       | <a href="#">Netkiller Security 手札</a>  | <a href="#">Netkiller Version 手札</a>      |
| <a href="#">Netkiller Intranet 手札</a>  | <a href="#">Netkiller Cisco IOS 手札</a>  | <a href="#">Netkiller Writer 手札</a>    | <a href="#">Netkiller Studio Linux 手札</a> |



### 目录

#### [自述](#)

- [1. 内容简介](#)
  - [1.1. Audience\(读者对象\)](#)
  - [1.2. 写给读者](#)
  - [1.3. 获得文档](#)
    - [1.3.1. PDF](#)
    - [1.3.2. EPUB](#)
    - [1.3.3. 获得光盘介质](#)

[2. 作者简介](#)

[2.1. 联系作者](#)

[3. 支持这个项目\(Support this project\)](#)

[I. Multi-dimension Architecture](#)

[1. Architecture Overview](#)

[1. CDN \(Content Delivery Network\)](#)

[2. Traffic](#)

[2.1. bandwidth](#)

[2.1.1. firewall](#)

[2.1.2. switch](#)

[2.1.3. Server](#)

[2.2. Session](#)

[2.2.1. firewall](#)

[2.2.2. Server](#)

[2.2.3. application](#)

[2.3. IO](#)

[2.3.1. 分布IO](#)

[2.3.2. FC SAN](#)

[2.3.3. iSCSI / FCoE](#)

[2.3.4. InfiniBand 或 RDMA](#)

[3. Distributed File System\(簇文件系统\)](#)

[3.1. 聚合文件系统](#)

[3.2. 全局文件系统](#)

[3.3. 负载均衡文件系统](#)

[3.4. 网络块设备](#)

[2. Systems architecture\(系统架构\)](#)

[1. 集群\(Cluster\)](#)

[1.1. 负载均衡](#)

[1.1.1. DNS负载均衡](#)

[1.1.2. 软件四层交换负载均衡](#)

[1.1.3. 应用层负载均衡](#)

[1.2. 高可用性集群](#)

[1.3. 负载均衡设备](#)

[1.4. 会话保持](#)

[1.5. 健康状态检查](#)

[2. 缓存技术](#)

[2.1. 浏览器缓存](#)

[2.1.1. Expires](#)

[2.1.2. If-Modified-Since / Last-Modified](#)

[2.1.3. ETag / If-None-Match](#)

[2.2. CDN/反向代理缓存](#)

[2.3. 负载均衡设备](#)

[2.4. WEB服务器缓存](#)

[2.5. 应用程序缓存](#)

[2.6. 数据库缓存](#)

[3. 静态化](#)

- [3.1. 生成方式](#)
- [3.2. 抓取方式](#)
- [3.3. 伪静态化](#)
- [3.4. 混合方式](#)
- [3.5. 静态化中的动态内容](#)

#### [4. 多媒体数据分离](#)

- [4.1. 图片服务器分离](#)
- [4.2. 目录层次规划](#)
- [4.3. 多域名访问](#)

#### [5. 压缩数据传输](#)

#### [6. 时间同步](#)

#### [7. 邮件系统](#)

##### [7.1. Mailing List](#)

#### [8. 日志集中管理](#)

- [8.1. 系统日志](#)
- [8.2. 应用程序日志](#)

#### [9. SSL](#)

#### [10. Storage 存储](#)

##### [10.1. 存储种类](#)

- [10.1.1. Direct Attached Storage](#)
- [10.1.2. Network-attached storage](#)
- [10.1.3. Storage area network](#)

###### [10.1.3.1. FC SAN](#)

###### [10.1.3.2. IP SAN](#)

###### [10.1.3.3. FCoE \(Fibre Channel over Ethernet\)](#)

##### [10.2. RAID](#)

- [10.2.1. 缓存服务器](#)
- [10.2.2. Web 服务器](#)
- [10.2.3. 数据库](#)
- [10.2.4. 数据备份](#)

##### [10.3. File System 文件系统](#)

###### [10.3.1. Distributed File System\(DFS\)](#)

##### [10.4. 数据访问协议](#)

##### [10.5. 数据管理](#)

- [10.5.1. Share 共享](#)
- [10.5.2. Mirror 远程镜像同步](#)
- [10.5.3. 压缩与重复数据消除](#)
- [10.5.4. Backup 备份与恢复](#)
- [10.5.5. 故障报告](#)

### [3. Software architecture \(软件架构\)](#)

#### [1. 开发框架 Framework](#)

- [1.1. HMVC](#)
- [1.2. REST](#)
- [1.3. SNA \(Shared Nothing Architecture\)](#)
- [1.4. 其他](#)

#### [2. URL设计](#)

- [2.1. URL 作为MVC 的Controller](#)
- [2.2. URL 伪静态化，用于SEO优化](#)

### [3. Javascript Framework](#)

#### [3.1. Minify JS](#)

##### [3.1.1. jsmin](#)

##### [3.1.2. yuicompressor](#)

##### [3.1.3. shrinksafe](#)

### [4. 模板\(template\)](#)

#### [4.1. HTML 页面优化](#)

### [5. Session/Cookie](#)

#### [5.1. Session](#)

#### [5.2. Session 共享](#)

#### [5.3. Cookie](#)

#### [5.4. Cookie 安全](#)

### [6. 国际化 Locale database。](#)

#### [6.1. Unicode](#)

#### [6.2. 繁简转换](#)

### [7. 数据库访问](#)

#### [7.1. CRUD](#)

#### [7.2. Active Record](#)

#### [7.3. OR Mapping](#)

### [8. Cache](#)

#### [8.1. 页面缓存](#)

#### [8.2. 局部缓存](#)

### [9. Single sign-on \(SSO\) 单点登录](#)

### [10. 搜索引擎](#)

### [11. Synchronous/Asynchronous](#)

### [12. Message Queuing](#)

### [13. Hash](#)

### [14. Sharding 垂直/水平切割](#)

#### [14.1. 面向服务](#)

#### [14.2. 面向数据库](#)

### [15. 日志系统](#)

## [II. Development](#)

### [4. 开源模式](#)

### [5. 开发语言及平台](#)

#### [1. 分层架构](#)

##### [1.1. 中间件 Middleware](#)

##### [1.2. 分层](#)

#### [2. Web 2.0](#)

#### [3. 云计算](#)

#### [4. 跨平台](#)

#### [5. 编译语言比脚本语言安全](#)

#### [6. 封装重用](#)

#### [7. 相关的工具](#)

##### [7.1. 开发工具](#)

### [6. project](#)

#### [1. Project directory](#)

[7. Logging](#)

[1. Remote Syslog](#)

[8. RDBMS 数据库设计](#)

[1. 数据字典](#)

[2. 用户帐号表](#)

[3. 分类表设计](#)

[3.1. 树形分类表](#)

[3.2. 多对多分类](#)

[3.3. 快速检索子分类设计](#)

[4. 文章表设计](#)

[4.1. 分区表设计](#)

[4.2. Title性能优化](#)

[5. 评论表](#)

[6. 记录点击率，阅读次数，及评分表](#)

[7. 产品属性表](#)

[7.1. 简单实现](#)

[7.2. 实现属性组管理](#)

[7.3. 可编辑属表](#)

[8. 国际化语言表](#)

[9. Workflow](#)

[10. 内容版本控制](#)

[11. Sharding](#)

[11.1. horizontal](#)

[11.2. vertical](#)

[9. 页面设计 Design](#)

[1. XHTML+DIV+CSS](#)

[2. 页面结构设计](#)

[2.1. Home page \(首页\)](#)

[2.2. 导航栏](#)

[2.3. Left Bar](#)

[2.4. 区块设计 Block](#)

[3. 表格](#)

[4. 图片优化](#)

[4.1. onmouseover/onmouseout](#)

[4.2. 使用一幅图片处理BLOCK四角](#)

[4.3. 图片用背景图代替img 标记](#)

[5. HTML嵌入图片](#)

[6. 页面内容安全](#)

[6.1. 禁止鼠标右键](#)

[6.2. 禁止复制剪切 及粘贴](#)

[7. html,css 有效性检查 Validation](#)

[10. Framework Design](#)

[1. MVC Framework](#)

[2. HMVC Framework](#)

[3. REST](#)

[3.1. RESTful JSON API](#)

[4. action 分发器](#)

---

5. [Plugin & Hook 设计与实现](#)

- [5.1. 插件管理平台](#)
- [5.2. 接口定义](#)
- [5.3. 插件](#)
- [5.4. 测试](#)

6. [Cache](#)

6.1. [CDN \(Content Delivery Network\) 与反向代理](#)

- [6.1.1. CDN接口API](#)
- [6.1.2. 方向代理页面过期处理](#)
- [6.1.3. 内容版本化](#)

6.2. [HTTP Cache](#)

- [6.2.1. 在程序中灵活操作 Cache-Control](#)
- [6.2.2. 非程序文件缓存处理](#)

6.3. [Cache 生存时间](#)

7. [Session/Cookie 安全](#)

7.1. [cookie-free domains](#)

8. [i18n 国际化](#)

- [8.1. 数组方式](#)
- [8.2. 数据库方式](#)
- [8.3. 文件文件](#)
- [8.4. Gettext](#)
- [8.5. 数据结构](#)

9. [Logging](#)

10. [SEO Module](#)

10.1. [robots.txt](#)

10.1.1. [Sitemap in robots.txt](#)

10.2. [sitemaps](#)

13. [Application Design](#)

- [1. Product](#)
- [2. Cart & Checkout](#)
- [3. Other](#)

[III. Operations](#)

14. [How Opensource Helps Optimize IT and Save You Money](#)

15. [运维管理管理](#)

- [1. 合同管理](#)
- [2. IDC 设备管理](#)

- [2.1. 资产管理](#)
- [2.2. 图表管理](#)

- [2.2.1. 网络拓扑图](#)
- [2.2.2. 机柜图](#)
- [2.2.3. 防火墙/路由器与交换机链路图](#)
- [2.2.4. 服务器连接图](#)

- [2.3. 进出IDC机房](#)
- [2.4. 设备上下架](#)
- [2.5. 如何维护设备](#)

16. [IDC](#)

[1. 接入线路](#)

[1.1. 带宽计算](#)

[1.2. 双线/多线](#)

[1.2.1. 智能DNS \(Smart DNS\)](#)

[1.3. BGP](#)

[1.4. 接入线路测试](#)

[2. 机柜\(Cabinets\)](#)

[2.1. 强电相关](#)

[2.2. IP 分配](#)

[3. Network](#)

[3.1. Firewall](#)

[3.1.1. DIY Firewall & VPN](#)

[3.1.1.1. Firewall](#)

[3.1.1.2. 3 Layer VPN](#)

[3.1.1.3. 7 Layer VPN](#)

[3.2. Switch](#)

[3.3. 物理拓扑图](#)

[4. 服务器部署与网络拓扑](#)

[4.1. 小型网站](#)

[4.2. 集群网站](#)

[4.3. 聚合端口](#)

[4.4. 关于服务器远程管理](#)

[5. RALDRS](#)

[17. Hardware](#)

[1. Server](#)

[1.1. TPC](#)

[1.2. IOPS \(Input/Output Operations Per Second, pronounced i-ops\)](#)

[1.3. rPerf](#)

[1.4. 网卡](#)

[1.4.1. 网口规划](#)

[1.4.1.1. 内外隔离](#)

[1.4.1.2. 负载均衡](#)

[1.4.1.3. 交叉互联](#)

[1.4.2. 常见网络适配器品牌](#)

[1.4.3. 1G 千兆以太网产品](#)

[1.4.4. 10G 万兆以太网产品](#)

[1.5. Fibre Channel HBA 卡](#)

[1.6. iDRAC / iLO / IMM](#)

[1.7. Hardware configuration](#)

[1.7.1. www/app](#)

[1.7.2. cache](#)

[1.7.3. database](#)

[1.8. 密码管理](#)

[2. Network](#)

[2.1. Firewall](#)

[2.1.1. Cisco ASA](#)

[2.1.2. Juniper](#)

[2.2. Switch](#)

[2.2.1. 2960G / 2960S](#)

[2.2.2. 3560 / 3750](#)

[2.2.3. 4948](#)

[2.2.4. 4506 / 4507](#)

[2.3. SLB](#)

[18. Linux Server](#)

[1. 服务器快速安装与部署](#)

[1.1. Kickstart](#)

[1.2. 快速初始化](#)

[1.3. JDK and Tomcat](#)

[1.4. 批量安装](#)

[2. Kernel](#)

[2.1. RPS / RFS](#)

[3. IP / Mac Address](#)

[3.1. IP Address](#)

[3.2. Mac Address](#)

[3.3. NIC](#)

[3.4. DNS](#)

[3.5. bonding](#)

[4. limits.conf](#)

[5. sysctl](#)

[6. profile](#)

[7. Crontab](#)

[8. NFS](#)

[9. 释放内存](#)

[10. OpenSSH](#)

[10.1. history 安全](#)

[10.2. sshd\\_config](#)

[11. TCP/IP](#)

[11.1. TCP 状态](#)

[11.2. TIME\\_WAIT](#)

[11.3. SYN\\_RECV / SYN\\_SENT](#)

[12. NTPD](#)

[13. dnsmasq](#)

[14. snmp](#)

[15. nrpe](#)

[16. vsftpd](#)

[17. 安装 LAMP](#)

[17.1. Apache](#)

[17.2. MySQL](#)

[17.3. PHP](#)

[17.4. resin-pro](#)

[17.5. httpd](#)

[17.6. php](#)

[17.7. httpd-2.2.21 + php-5.3.8 自动安装脚本](#)

[18. Linux 日常管理](#)

[20. Monitor solution](#)



## [1. 网络监控](#)

### [1.1. 流量监控](#)

### [1.2. 交换机监控](#)

## [2. 集群监控](#)

## [3. 服务器状态监控](#)

### [3.1. 负载监控](#)

### [3.2. 磁盘空间监控](#)

### [3.3. 内存监控](#)

### [3.4. CPU监控](#)

## [4. MySQL 监控](#)

## [5. 网站安全与预警机制](#)

## [21. Backup](#)

### [1. help](#)

#### [1.1. Task](#)

#### [1.2. Schedule](#)

#### [1.3. Crontab](#)

## [2. 配置文件备份](#)

### [2.1. Firewall and Switch](#)

### [2.2. Server](#)

## [22. 常用工具](#)

### [1. IP Zone](#)

### [2. What is my IP address?](#)

#### [2.1.](#)

### [3. IP Tracking](#)

### [4. 自动化脚本](#)

#### [4.1. Install Script](#)

#### [4.2. FTP User Script](#)

#### [4.3. SVN User Script](#)

## [23. 服务器安全](#)

### [1. 防火墙](#)

### [2. Web Server](#)

#### [2.1. 目录访问控制](#)

#### [2.2. fastcgi 安全](#)

#### [2.3. php 函数安全](#)

#### [2.4. 服务器版本信息](#)

## [24. 运维自动化](#)

### [1. Cobbler](#)

### [2. Cfengine](#)

### [3. Puppet](#)

### [4. func](#)

## [IV. SQA \(Software Quality Assurance\)](#)

## [25. SCM \(Software Configuration Management\)](#)

## [26. Automatic Testing](#)

### [1. 静态内容并发测试](#)

### [2. 动态内容测试](#)

### [3. 业务逻辑并发能力](#)

### [4. Cluster Testing](#)

## [27. Software Testing](#)

### [1. 缺陷管理工具](#)

#### [1.1. Mantis](#)

##### [1.1.1. JpGraph](#)

#### [1.2. Bugzilla](#)

## [28. IO Testing](#)

### [1. fio](#)

### [2. iozone](#)

### [3. iometer](#)

### [4. 测试文件系统](#)

### [5. iperf: Tool for measuring TCP and UDP bandwidth performance](#)

## [29. 压力测试工具](#)

### [1. TCP/IP 压力测试优化](#)

### [2. ApacheBench](#)

### [3. JMeter](#)

### [4. Siege](#)

## [30. Test - System performance benchmark](#)

### [1. SysBench](#)

## [31. OpenSTA\(Open System Testing Architecture\)](#)

## [32. Web Testing](#)

### [1. Browser Config](#)

#### [1.1. Chrome](#)

### [2. WATIR - WEB AUTOMATION TESTING IN RUBY](#)

#### [2.1. Install of WATIR](#)

##### [2.1.1. IE](#)

##### [2.1.2. Firefox](#)

#### [2.2. Getting Started](#)

#### [2.3. Config file](#)

#### [2.4. Watir library](#)

#### [2.5. Browser](#)

##### [2.5.1. Environment variable](#)

##### [2.5.2. Browser options](#)

##### [2.5.3. Open a browser \(default: Internet Explorer\)](#)

##### [2.5.4. Watir::Browser.attach](#)

##### [2.5.5. Open Browser at the specified URL](#)

##### [2.5.6. Go to a specified URL](#)

##### [2.5.7. Close the browser](#)

#### [2.6. Test Unit](#)

#### [2.7. HTML Tags and Watir Methods](#)

##### [2.7.1. Collections of HTML Elements](#)

### [3. Selenium - Web Browser Automation](#)

#### [3.1. ruby](#)

#### [3.2. python](#)

### [4. WEBDRIVER](#)

### [5. WatiN](#)

### [6. PAMIE](#)

[33. What is DevOps](#)

- [1. Trac Project Management](#)
- [2. Development Environment](#)
- [3. Testing Environment](#)
- [4. Coding workflow](#)
- [5. Automation Deployment](#)

[5.1. Deployment Toolkit](#)

[5.2. Deployment Script](#)

[5.2.1. FTP Script](#)

[5.2.2. SFTP Script](#)

[5.2.3. SCP Script](#)

[6. Upgrade](#)

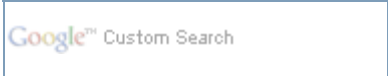
[A. 附录](#)  
[术语表](#)

范例清单

- 9.1. [表格排版范例](#)
- 9.2. [XHTML+DIV+CSS排版范例](#)
- 9.3. [例子](#)
- 9.4. [例子](#)
- 10.1. [php language package](#)
- 10.2. [sql table language package](#)
- 10.3. [example robots.txt](#)
- 12.1. [Example for ECSHOP](#)
- 21.1. [Backup program](#)

修订历史

修订 0.1.1	Sep 12, 2011	Neo
章节做了大调整，将文档分为五块，多维度架构，开发，运维，SQA，还有DevOps。		
修订 0.1.0	May 15, 2010	Neo
增加解决方案一节，并填充了大量章节。同时对完成这篇文档信心大增		
修订 0.0.4	2010	Neo
这篇文档几乎没有时间和精力编辑，内容增加不多。		
修订 0.0.4	April 15, 2009	Neo
这篇文档几乎搁浅，没有时间和精力，没有编辑加入。今天做了一下布局调整，增加一些内容。		
修订 0.0.3	Sep. 17, 2008	Neo
加入关于存储的内容		
修订 0.0.1	May 24, 2008	Neo
李振韬加入编译团队		
修订 0.0.0	May 22, 2008	Neo
这是一个值得纪念的日子		



# 自述

## 目录

### [1. 内容简介](#)

#### [1.1. Audience\(读者对象\)](#)

#### [1.2. 写给读者](#)

#### [1.3. 获得文档](#)

##### [1.3.1. PDF](#)

##### [1.3.2. EPUB](#)

##### [1.3.3. 获得光盘介质](#)

### [2. 作者简介](#)

#### [2.1. 联系作者](#)

### [3. 支持这个项目 \(Support this project\)](#)

## 1. 内容简介

当前文档档容比较杂，涉及内容广泛。

慢慢我会将其中章节拆成新文档.

文档内容简介:

1. Network
2. Security
3. Web Application
4. Database
5. Storage And Backup/Restore
6. Cluster
7. Developer

### 1.1. Audience(读者对象)

This book is intended primarily for Linux system administrators who are familiar with the following activities:

Audience

1. Linux system administration procedures, including kernel configuration
2. Installation and configuration of cluster, such as load balancing, High Availability,
3. Installation and configuration of shared storage networks, such as Fibre Channel SANs
4. Installation and configuration of web server, such as apache, nginx, lighttpd, tomcat/resin ...

本文档的读者对象:

文档面向有所有读者。您可以选读您所需要的章节,无需全篇阅读,因为有些章节不一定对你有用,用得着就翻来看看,暂时用不到的可以不看.

大体分来读者可以分为几类:

1. 架构工程师
2. 系统管理员
3. 系统支持,部署工程师

不管是谁,做什么的,我希望通过阅读这篇文档都能对你有所帮助。

## 1.2. 写给读者

欢迎提出宝贵的建议,如有问题请到 [邮件列表](#) 讨论

为什么写这篇文章

有很多想法,工作中也用不到所以未能实现,所以想写出来,和大家分享.有一点写一点,写得也不好,只要能看懂就行,就当学习笔记了.

开始零零碎碎写过一些文档,也向维基百科供过稿,但维基经常被ZF封锁,后来发现sf.net可以提供主机存放文档,便做了迁移。并开始了我的写作生涯。

这篇文档是作者8年来对工作的总结,是作者一点一滴的积累起来的,有些笔记已经丢失,所以并不完整。

因为工作太忙整理比较缓慢。目前的工作涉及面比较窄所以新文档比较少。

我现在花在技术上的时间越来越少,兴趣转向摄影,无线电。也想写写摄影方面的心得体会。

写作动力:

曾经在网上看到外国开源界对中国的评价,中国人对开源索取无度,但贡献却微乎其微.这句话一直记在我心中,发誓要为中国开源事业做我仅有的一点微薄贡献

另外写文档也是知识积累,还可以增加在圈内的影响力.

人跟动物的不同,就是人类可以把自己学习的经验教给下一代人.下一代在上一代的基础上再创新,不断积累才有今天.

所以我把自己的经验写出来,可以让经验传承

没有内容的章节:

目前我自己一人维护所有文档,写作时间有限,当我发现一个好主题就会加入到文档中,待我有时间再完善章节,所以你会发现很多章节是空无内容的.

文档目前几乎是流水帐式的写作,维护量很大,先将就着看吧.

我想到哪写到哪,你会发现文章没一个中心,今天这里写点,明天跳过本章写其它的.

文中例子绝对多,对喜欢复制然后粘贴朋友很有用,不用动手写,也省时间.

理论的东西,网上大把,我这里就不写了,需要可以去网上查.

我爱写错别字,还有一些是打错的,如果发现请指正.

文中大部分试验是在Debian/Ubuntu/Redhat AS上完成.

## 1.3. 获得文档

### 1.3.1. PDF

[Download PDF Document](#) 下载PDF文档1

[Download PDF Document](#) 下载PDF文档2

### 1.3.2. EPUB

<http://netkiller.sourceforge.net/technology.html>

1.3.3. 获得光盘介质

如有特别需要，请联系我

## 2. 作者简介 自述

[上一页](#)

[下一页](#)

[Home](#) | [Mirror](#) | [Search](#)

Google™ Custom Search

## 2. 作者简介

主页地址: <http://netkiller.sourceforge.net>, <http://netkiller.github.com/>

陈景峰 (ネッカリムラタ)

Nickname: netkiller | English name: Neo chen | Nippon name: ちんけいほう (音訳) | Korean name: | Thailand name:

IT民工, UNIX like Evangelist, 业余无线电爱好者 (呼号: BG7NYT), 户外运动以及摄影爱好者。

《PostgreSQL实用实例参考》, 《Postfix 完整解决方案》, 《Netkiller Linux 手札》的作者  
2001年来深圳进城打工,成为一名外来务工者。

2002年我发现不能埋头苦干,埋头搞技术是不对的,还要学会"做人".

2003年这年最惨,公司拖欠工资16000元,打过两次官司2005才付清.

2004年开始加入 [分布式计算](#) 团队, [目前成绩](#)

2004-10月开始玩户外和摄影

2005-6月成为中国无线电运动协会会员

2006年单身生活了这么多年,终于找到归宿.

2007物价上涨,金融危机,休息了4个月 (其实是找不到工作)

2008终于找到英文学习方法, , 《Netkiller Developer 手札》, 《Netkiller Document 手札》

2008-8-8 08:08:08 结婚,后全家迁居湖南省常德市

2009 《Netkiller Database 手札》,年底拿到C1驾照

2010对电子打击乐产生兴趣,计划学习爵士鼓

2011 职业生涯路上继续打怪升级

### 2.1. 联系作者

Mobile: +86 13113668890

Tel: +86 755 2981-2080

Callsign: BG7NYT QTH: Shenzhen, China

注: 请不要问我安装问题!

E-Mail: [openunix@163.com](mailto:openunix@163.com)

IRC [#ubuntu](irc://irc.freenode.net) / [#ubuntu-cn](irc://irc.freenode.net)

Yahoo: [bg7nyt](#)

ICQ: 101888222

AIM: [bg7nyt](#)

TM/QQ: 13721218  
MSN: netkiller@msn.com  
G Talk: 很少开  
网易泡泡：很少开

写给火腿:

欢迎无线电爱好者和我QSO,我的QTH在深圳宝安区龙华镇溪山美地12B7CD,设备YAESU  
FT-50R,FT-60R,FT-7800 144-430双段机,拉杆天线/GP天线 Nagoya MAG-79EL-3W/Yagi

如果这篇文章对你有所帮助,请寄给我一张QSL卡片,[qrz.cn](#) or [qrz.com](#) or [hamcall.net](#)

Personal Amateur Radiostations of P.R.China

ZONE CQ24 ITU44 ShenZhen, China

Best Regards, VY 73! OP. BG7NYT

---

[上一页](#)  
自述

[上一级](#)  
[起始页](#)

[下一页](#)  
3. 支持这个项目(Support this project)





3. 支持这个项目(Support this project)

[Donations](#)

招商银行(China Merchants Bank) 陈景峰 9555500000007459



# 部分 I. Multi-dimension Architecture

Design and implementation of a website solution with step by step

目录

[1. Architecture Overview](#)

[1. CDN \(Content Delivery Network\)](#)

[2. Traffic](#)

[2.1. bandwidth](#)

[2.1.1. firewall](#)

[2.1.2. switch](#)

[2.1.3. Server](#)

[2.2. Session](#)

[2.2.1. firewall](#)

[2.2.2. Server](#)

[2.2.3. application](#)

[2.3. IO](#)

[2.3.1. 分布IO](#)

[2.3.2. FC SAN](#)

[2.3.3. iSCSI / FCoE](#)

[2.3.4. InfiniBand 或 RDMA](#)

[3. Distributed File System\(簇文件系统\)](#)

[3.1. 聚合文件系统](#)

[3.2. 全局文件系统](#)

[3.3. 负载均衡文件系统](#)

[3.4. 网络块设备](#)

[2. Systems architecture\(系统架构\)](#)

[1. 集群\(Cluster\)](#)

[1.1. 负载均衡](#)

[1.1.1. DNS负载均衡](#)

[1.1.2. 软件四层交换负载均衡](#)

[1.1.3. 应用层负载均衡](#)

[1.2. 高可用性集群](#)

[1.3. 负载均衡设备](#)

[1.4. 会话保持](#)

[1.5. 健康状态检查](#)

[2. 缓存技术](#)

[2.1. 浏览器缓存](#)

[2.1.1. Expires](#)

[2.1.2. If-Modified-Since / Last-Modified](#)

[2.1.3. ETag / If-None-Match](#)

[2.2. CDN/反向代理缓存](#)

[2.3. 负载均衡设备](#)

[2.4. WEB服务器缓存](#)

[2.5. 应用程序缓存](#)

[2.6. 数据库缓存](#)

[3. 静态化](#)

[3.1. 生成方式](#)

[3.2. 抓取方式](#)

[3.3. 伪静态化](#)

[3.4. 混合方式](#)

[3.5. 静态化中的动态内容](#)

[4. 多媒体数据分离](#)

[4.1. 图片服务器分离](#)

[4.2. 目录层次规划](#)

[4.3. 多域名访问](#)

[5. 压缩数据传输](#)

[6. 时间同步](#)

[7. 邮件系统](#)

[7.1. Mailing List](#)

[8. 日志集中管理](#)

[8.1. 系统日志](#)

[8.2. 应用程序日志](#)

[9. SSL](#)

[10. Storage 存储](#)

[10.1. 存储种类](#)

[10.1.1. Direct Attached Storage](#)

[10.1.2. Network-attached storage](#)

[10.1.3. Storage area network](#)

[10.1.3.1. FC SAN](#)

[10.1.3.2. IP SAN](#)

[10.1.3.3. FCoE \(Fibre Channel over Ethernet\)](#)

[10.2. RAID](#)

[10.2.1. 缓存服务器](#)

[10.2.2. Web 服务器](#)

[10.2.3. 数据库](#)

[10.2.4. 数据备份](#)

[10.3. File System 文件系统](#)

[10.3.1. Distributed File System\(DFS\)](#)

[10.4. 数据访问协议](#)

[10.5. 数据管理](#)

[10.5.1. Share 共享](#)

[10.5.2. Mirror 远程镜像同步](#)

[10.5.3. 压缩与重复数据消除](#)

[10.5.4. Backup 备份与恢复](#)

[10.5.5. 故障报告](#)

[3. Software architecture \(软件架构\)](#)

[1. 开发框架 Framework](#)

[1.1. HMVC](#)

[1.2. REST](#)

[1.3. SNA \(Shared Nothing Architecture\)](#)

[1.4. 其他](#)

## [2. URL设计](#)

[2.1. URL 作为MVC 的Controller](#)

[2.2. URL 伪静态化，用于SEO优化](#)

## [3. Javascript Framework](#)

[3.1. Minify JS](#)

[3.1.1. jsmin](#)

[3.1.2. yuicompressor](#)

[3.1.3. shrinksafe](#)

## [4. 模板\(template\)](#)

[4.1. HTML 页面优化](#)

## [5. Session/Cookie](#)

[5.1. Session](#)

[5.2. Session 共享](#)

[5.3. Cookie](#)

[5.4. Cookie 安全](#)

## [6. 国际化 Locale database。](#)

[6.1. Unicode](#)

[6.2. 繁简转换](#)

## [7. 数据库访问](#)

[7.1. CRUD](#)

[7.2. Active Record](#)

[7.3. OR Mapping](#)

## [8. Cache](#)

[8.1. 页面缓存](#)

[8.2. 局部缓存](#)

## [9. Single sign-on \(SSO\) 单点登录](#)

---

[10. 搜索引擎](#)

[11. Synchronous/Asynchronous](#)

[12. Message Queuing](#)

[13. Hash](#)

[14. Sharding 垂直/水平切割](#)

[14.1. 面向服务](#)

[14.2. 面向数据库](#)

[15. 日志系统](#)

---

[上一页](#)

[下一页](#)

3. 支持这个项目(Support this project)

[起始页](#)

第 1 章 Architecture Overview



# 第 1 章 Architecture Overview

目录

[1. CDN \(Content Delivery Network\)](#)

[2. Traffic](#)

[2.1. bandwidth](#)

[2.1.1. firewall](#)

[2.1.2. switch](#)

[2.1.3. Server](#)

[2.2. Session](#)

[2.2.1. firewall](#)

[2.2.2. Server](#)

[2.2.3. application](#)

[2.3. IO](#)

[2.3.1. 分布IO](#)

[2.3.2. FC SAN](#)

[2.3.3. iSCSI / FCoE](#)

[2.3.4. InfiniBand 或 RDMA](#)

[3. Distributed File System\(簇文件系统\)](#)

[3.1. 聚合文件系统](#)

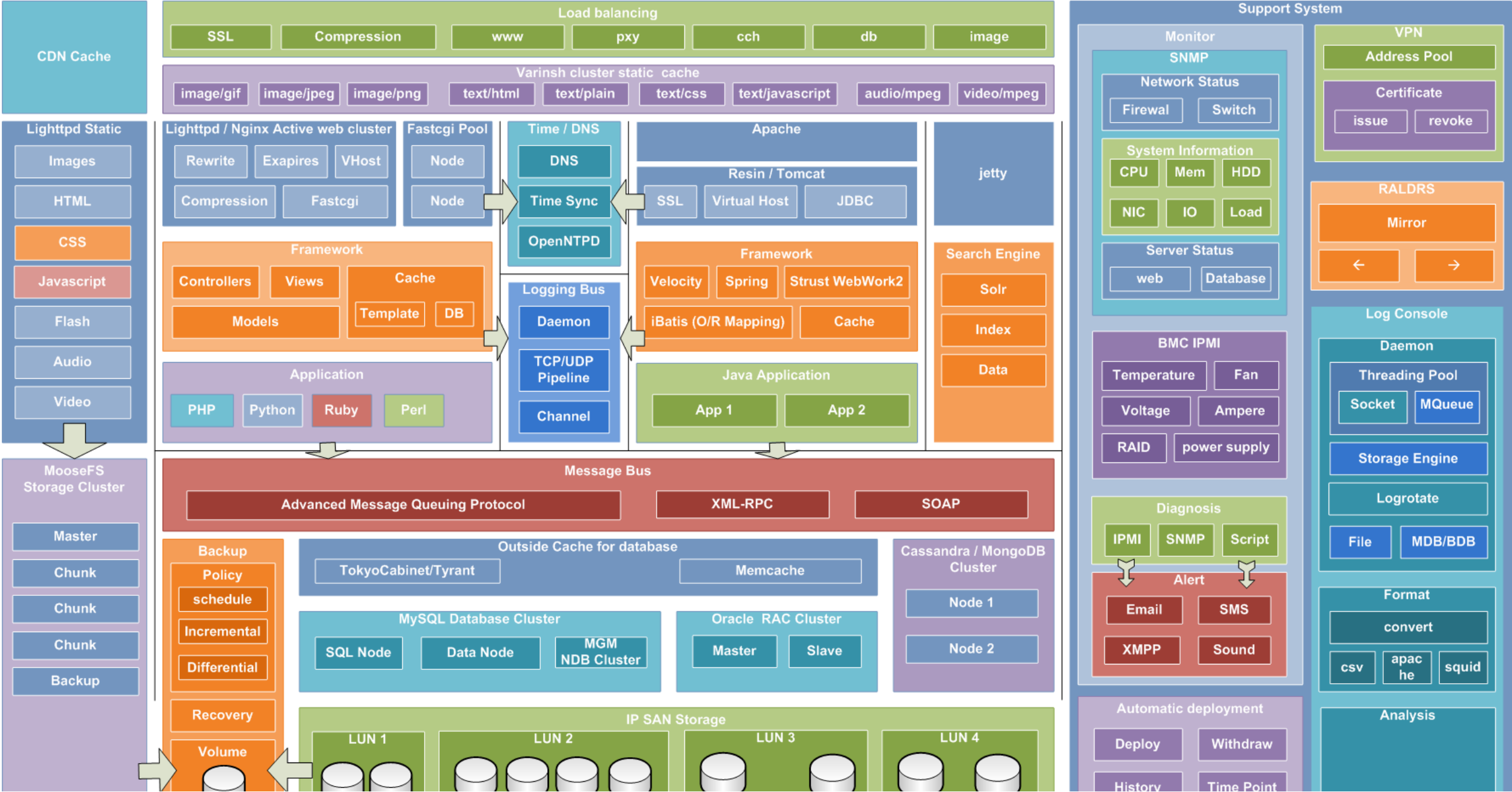
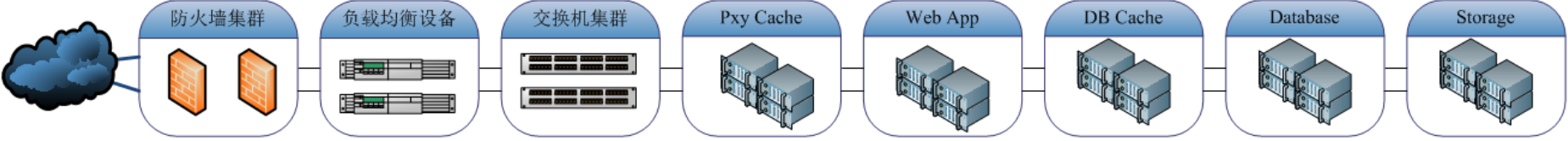
[3.2. 全局文件系统](#)

[3.3. 负载均衡文件系统](#)

[3.4. 网络块设备](#)

Design by Neo

Homepage:  
<http://netkiller.sourceforge.net/>  
<http://netkiller.github.com/>  
Author: netkiller <openunix@163.com>

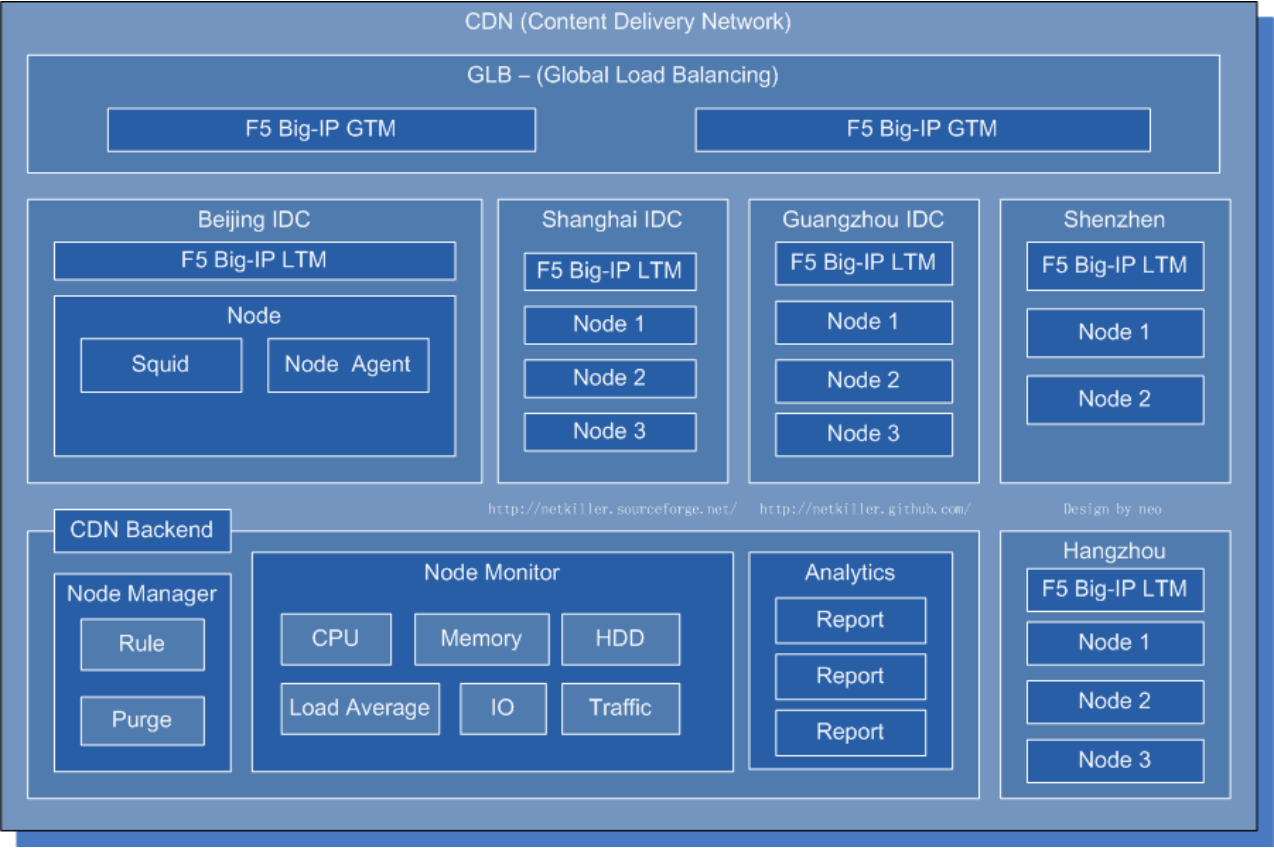






## 1. CDN (Content Delivery Network)

I analyzed CDN realization principle, look at the picture below.



[上一页](#)

部分 I. Multi-dimension Architecture

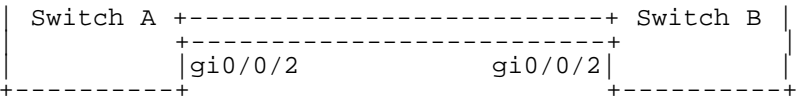
[上一级](#)

[起始页](#)

[下一页](#)

2. Traffic





比如你有5个机柜，将交换机放置到3号机柜，处于中间位置，所有交换机放入该机柜，然后堆叠，从中心机柜向两侧分线



对于不大不小的企业，直接采购IDC箱式交换机



2.1.3. Server

目前主流服务都配备2到4个网口，像IBM / HP / DELL 等品牌服务器你无需关心网卡问题.

这里主要是针对自行安装或使用PC服务器的用户，因为很多PC服务器使用Realtek网卡。那么Realtek与Broadcom的NetXtreme有什么不同？

建议你安装一个windows系统在服务器上，然后看看网卡驱动属性。Realtek 仅提供基本网络功能，QOS质量访问服务由驱动程序提供（软QOS）而NetXtreme 提供非常丰富的功能，并且都是硬件实现。

话题回到带宽上，linux 支持 bonding 网卡，可以帮你解决服务器网络通信带宽问题，bonding 还可以解决网卡故障转移，传输流量负载均衡等等。

在我的《Netkiller Linux 手札》中你可以找到具体的设置方法。

## 2.2. Session

firewall (nat session) - switch (Forwarding bandwidth) - os (ulimit,sysctl) - application (httpd,vsftpd,tomcat ...)

会话数，国人俗称并发数。当你的带宽没有满，但tcp不能建立连接，这时你就要考虑会话数了。

### 2.2.1. firewall

购买防火墙的时候主要有两个指标，一是会话数，二是带宽，三是配备模块。售前工程师都会交代清楚。

例如 Cisco ASA 5550 会话数65万，2个1G接口，可选IPS模块等等...

使用下面命令可以查看当前会话数

```
show conn count
```

### 2.2.2. Server

Linux 影响会话数的的参数与配置文件

/etc/security/limits.conf , /etc/security/limits.d

nofile - max number of open files 在POSIX系统中硬件，管道，Socket 均被看作是一个设备，如硬盘是块设备，显示器是字符设备，操作这些设备均使用c的open函数，被算作打开一个文件。所有设备都是如此，加上web服务器还要读取很多HTML文件，系统对nofile 开销是非常巨大的。

nproc - max number of processes 目前多线程是主流，使用多线程技术这个参数可以不关心。像Oracle,PostgreSQL, Apache prefork,你就需要关心这个参数

/etc/sysctl.conf , /etc/sysctl.d/

net.ipv4.ip\_local\_port\_range = 1024 65500 可用端口范围

tcp 协议当你尝试主动与服务器建立连接，如：telnet 172.16.0.1 80,本地会开启一个大于1024小于65500的端口

client: localhost:1025 --- 172.16.0.1:80 server

以上参数要综合你的CPU处理能力，内存空间，硬盘IO等等，才能配置出合理数值

配置过大（小马拉大车），超出你的服务器处理能力，导致服务器无响应，最终只能重启

配置过小（大马拉小车），你的服务器长时间处于空闲状态，CPU，内存没有得到合理使用

在我的《Netkiller Linux 手札》中你可以找到具体的设置方法。

### 2.2.3. application

连接数受限与limits与sysctl

apache : httpd/conf/extra/httpd-mpm.conf

```
<IfModule mpm_worker_module>
    ServerLimit          16
    ThreadLimit          256
    StartServers         8
    MaxClients           4096
    MinSpareThreads      64
    MaxSpareThreads      256
    ThreadsPerChild      256
    MaxRequestsPerChild  10000
</IfModule>
```

mysql : /etc/my.cnf

```
[mysqld]
max_connections=250
```

不依依列举，有兴趣看我的系列文档。

## 2.3. IO

IO (Input/Output) 输入/输出，在国内被泛指硬盘IO，没办法这里也不例外，也被指为硬盘IO

影响IO的几个参数：

硬盘转速与硬盘速率

RAID卡速率

以Dell为例，去官网查看一下

[http://www.dell.com/content/topics/topic.aspx/global/products/pvaul/topics/en/us/raid\\_controller?c=us&l=en&cs=555](http://www.dell.com/content/topics/topic.aspx/global/products/pvaul/topics/en/us/raid_controller?c=us&l=en&cs=555)

PERC H700 Integrated / Adapter: 6Gb/s SAS

SAS 硬盘接口 3Gbps，理论读写速度300MB/S，实际情况没有这么理想。

RAID0 / RAID10是提高IO最有效的手段，但是你从上面数据计算。6块SAS硬盘做Raid 0 传输速率可以达到18Gb/s，但RAID卡H700只能达到6Gb/s，整体带宽并没有提高。

这样做的意义是在Raid带宽与硬盘速度不变的情况下，读写所花费的时间减少了，提高了列队处理速度，减少IO排队。

IO的问题就是IO排队等待问题，而不是传输带宽不够用

### 2.3.1. 分布IO

在经济紧张的情况下，可以使用多块独立硬盘分布IO，每块硬件分别做独立存储，比如数据库可以采用这种方案：可以一块硬盘存数据，一块硬盘做索引，另一块做日志等等，禁止交叉。

在经济允许的情况下，你可以配置多个RAID卡，外挂DAS。或者采用集群加分布式文件系统方案

2.3.2. FC SAN

8Gb Fibre Channel

我曾经测试过本地硬盘（146G 15RPM \* 8 做RAID10）

2.3.3. iSCSI / FCoE

<http://zh.wikipedia.org/wiki/ISCSI>

iSCSI 可以提供1GB，10GB数据传输，传输介质可以选择双绞线或者光纤

FCoE 通过以太网传输FC协议，与iSCSI有很多相似之处

2.3.4. InfiniBand 或 RDMA

提供10Gbps ~ 120Gbps 的IO速度

<http://en.wikipedia.org/wiki/InfiniBand>

<http://www.infinibandta.org/>



### 3. Distributed File System(簇文件系统)

我把分布式文件系统分为三类，聚合文件系统，全局文件系统，负载均衡文件系统。

除了gfs其他文件系统都是建立在本地文件系统之上的网络文件系统。

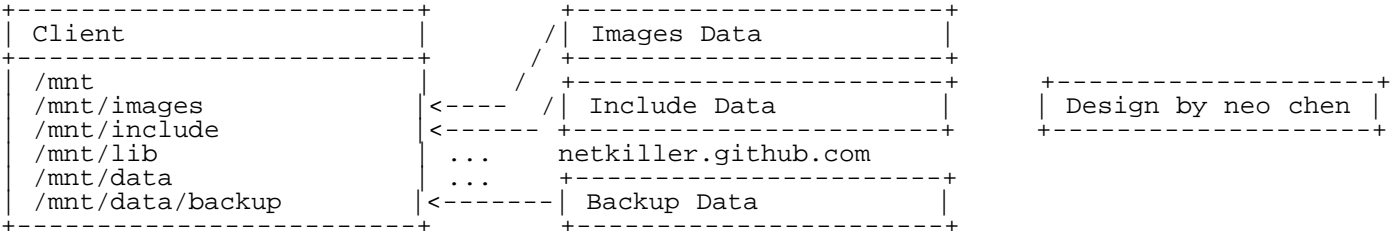
几乎所有DFS都能通过fuse mount 到本地，但有些DFS mount 后性能不佳。

还有一个与分布式文件系统密切相关的，就是块设备，块设备不是文件系统，可以称为裸设备。

#### 3.1. 聚合文件系统

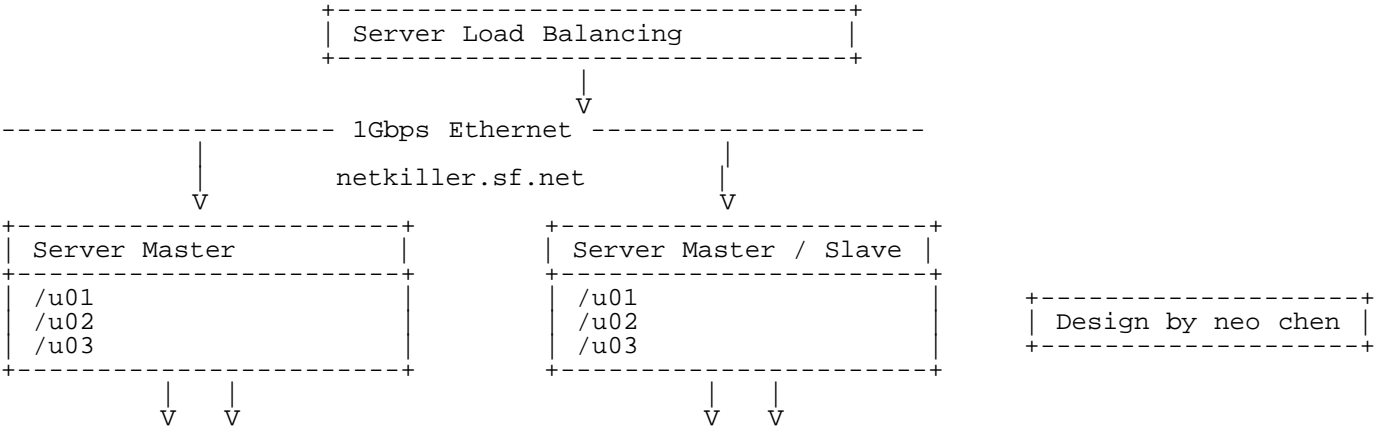
以NFS, glusterfs 为代表，其特点是server独立运行，Server与Server间没有通信，然后访问者将其聚合组织并规划目录，为client提供数据共享。

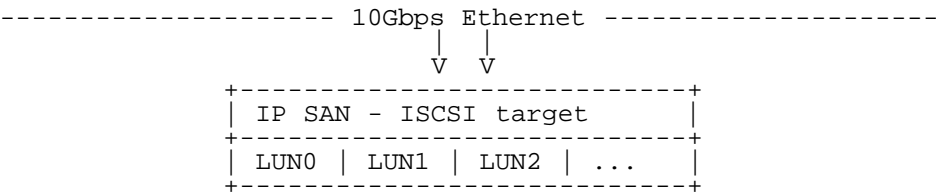
glusterfs 可以实现Mirror与Strip等更复杂的组合，但全由client完成，server之间没有交互。



#### 3.2. 全局文件系统

如 gfs，它可以提供server间文件系统协商，同步元数据等等。常规文件系统只能用于本地硬盘，如果两个服务器同时mount iscsi存储，会出现A服务器写入后，B服务器无法看到A刚刚写入的数据，如果两台同时写入数据，会损坏文件系统。





3.3. 负载均衡文件系统

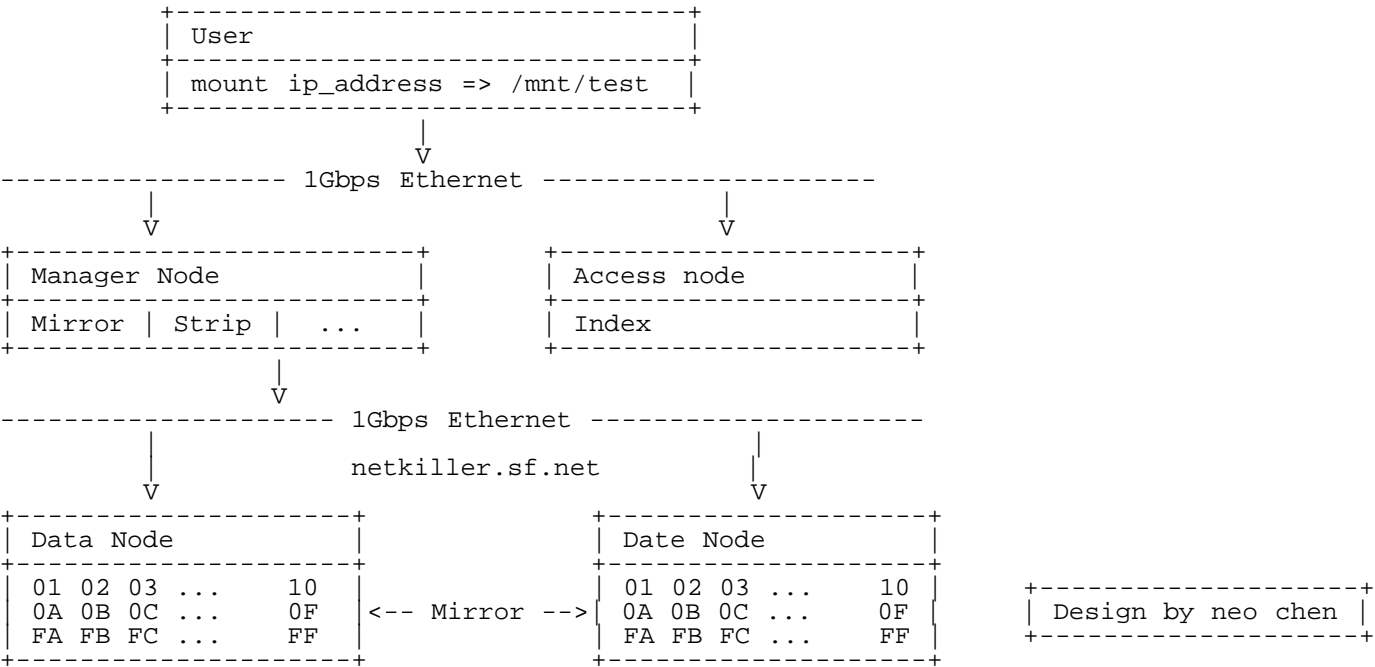
这种文件系统通常至少有三部分组成，存储节点，访问节点，管理节点。不同的系统叫法不同，但其原理相同。

存储节点,负责数据存储，数据通过hash散列

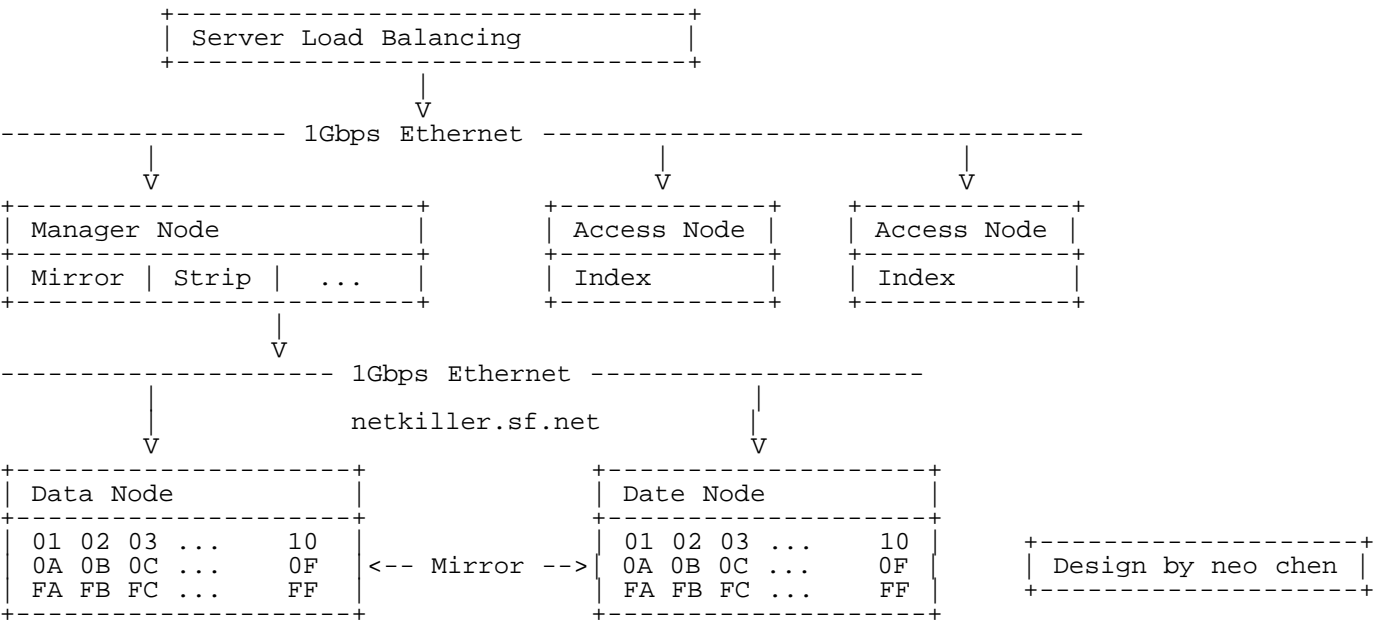
访问节点，用户通过该节点访问数据，做数据上传下载。访问方式分为点对点与三角方式

管理节点，服务数据Mirror,Strip等，元数据同步等等...

点到点系统只提供一个访问入口，如： MooseFS



三角链路





这种文件系统的特点是，当用户访问文件系统时，首先访问管理节点，管理节点会返回一个数据地址，用户再从访问节点的地址取得数据。

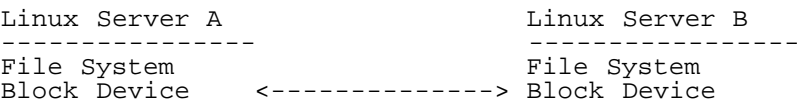
以MogileFS为代表

某些系统甚至直接使用反向代理或者WEB服务器作为访问节点。这种系统非常适合多媒体数据存储。通过负载均衡可能实现横向与纵向灵活扩展

### 3.4. 网络块设备

本地文件系统是建立在块设备之上的。使用块设备，首先配置好块设备，然后你就可以把它当成物理硬盘一样对待，在块设备上分区，格式化。

以DRBD,nbd-server为代表，网络块设备可以保证两块物理硬盘的数据同步，常用语HA集群



更多细节参考 <http://netkiller.github.com/storage/>



## 第 2 章 Systems architecture(系统架构)

### Systems architecture

#### 目录

#### [1. 集群\(Cluster\)](#)

##### [1.1. 负载均衡](#)

###### [1.1.1. DNS负载均衡](#)

###### [1.1.2. 软件四层交换负载均衡](#)

###### [1.1.3. 应用层负载均衡](#)

##### [1.2. 高可用性集群](#)

##### [1.3. 负载均衡设备](#)

##### [1.4. 会话保持](#)

##### [1.5. 健康状态检查](#)

#### [2. 缓存技术](#)

##### [2.1. 浏览器缓存](#)

###### [2.1.1. Expires](#)

###### [2.1.2. If-Modified-Since / Last-Modified](#)

###### [2.1.3. ETag / If-None-Match](#)

##### [2.2. CDN/反向代理缓存](#)

##### [2.3. 负载均衡设备](#)

##### [2.4. WEB服务器缓存](#)

##### [2.5. 应用程序缓存](#)

##### [2.6. 数据库缓存](#)

#### [3. 静态化](#)

##### [3.1. 生成方式](#)

##### [3.2. 抓取方式](#)

##### [3.3. 伪静态化](#)

[3.4. 混合方式](#)

[3.5. 静态化中的动态内容](#)

[4. 多媒体数据分离](#)

[4.1. 图片服务器分离](#)

[4.2. 目录层次规划](#)

[4.3. 多域名访问](#)

[5. 压缩数据传输](#)

[6. 时间同步](#)

[7. 邮件系统](#)

[7.1. Mailing List](#)

[8. 日志集中管理](#)

[8.1. 系统日志](#)

[8.2. 应用程序日志](#)

[9. SSL](#)

[10. Storage 存储](#)

[10.1. 存储种类](#)

[10.1.1. Direct Attached Storage](#)

[10.1.2. Network-attached storage](#)

[10.1.3. Storage area network](#)

[10.1.3.1. FC SAN](#)

[10.1.3.2. IP SAN](#)

[10.1.3.3. FCoE \(Fibre Channel over Ethernet\)](#)

[10.2. RAID](#)

[10.2.1. 缓存服务器](#)

[10.2.2. Web 服务器](#)

[10.2.3. 数据库](#)

[10.2.4. 数据备份](#)

[10.3. File System 文件系统](#)

[10.3.1. Distributed File System\(DFS\)](#)

[10.4. 数据访问协议](#)

[10.5. 数据管理](#)

[10.5.1. Share 共享](#)

[10.5.2. Mirror 远程镜像同步](#)

[10.5.3. 压缩与重复数据消除](#)

[10.5.4. Backup 备份与恢复](#)

[10.5.5. 故障报告](#)

# 1. 集群(Cluster)

集群有很多实现方法，分为硬件和软件，集群可以在不同网络层面上实现

- 1. 实现IP轮循（Bind DNS）
- 2. 硬件四层交换（硬件负载均衡设备 F5 BIG IP）
- 3. 软件四层交换（linux virtual server）
- 4. 应用层上实现（tomcat）

越是低层性能越好，越是上层功能更强

集群的分类

- 1. 高可用性集群
- 2. 负载均衡集群
- 3. 超级计算集群

网站一般用到两种集群分别是高可用性集群和负载均衡集群

## 1.1. 负载均衡

### 1.1.1. DNS负载均衡

这是早期出现的负载均衡技术，直到现在，很多网站仍然使用DNS负载均衡。

你可通过ping命令观看它是如何工作的，例如你可反复ping个网域名。

```
C:\>ping www.163.com

Pinging www.cache.split.netease.com [220.181.28.52] with 32 bytes of data:

Reply from 220.181.28.52: bytes=32 time=226ms TTL=53
Reply from 220.181.28.52: bytes=32 time=225ms TTL=53
Reply from 220.181.28.52: bytes=32 time=226ms TTL=53
Reply from 220.181.28.52: bytes=32 time=226ms TTL=53

Ping statistics for 220.181.28.52:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 225ms, Maximum = 226ms, Average = 225ms

C:\>ping www.163.com

Pinging www.cache.split.netease.com [220.181.28.53] with 32 bytes of data:
```

```
Reply from 220.181.28.53: bytes=32 time=52ms TTL=52
Reply from 220.181.28.53: bytes=32 time=53ms TTL=52
Reply from 220.181.28.53: bytes=32 time=52ms TTL=52
Reply from 220.181.28.53: bytes=32 time=52ms TTL=52

Ping statistics for 220.181.28.53:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 52ms, Maximum = 53ms, Average = 52ms

C:\>ping www.163.com

Pinging www.cache.split.netease.com [220.181.28.50] with 32 bytes of data:

Reply from 220.181.28.50: bytes=32 time=51ms TTL=53
Reply from 220.181.28.50: bytes=32 time=52ms TTL=53
Reply from 220.181.28.50: bytes=32 time=52ms TTL=53
Reply from 220.181.28.50: bytes=32 time=51ms TTL=53

Ping statistics for 220.181.28.50:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 51ms, Maximum = 52ms, Average = 51ms

C:\>
```

DNS负载均衡主要优点

- 1. 技术简单，容易实现，灵活，方便，成本低
- 2. Web服务器可以位于互联网的任意位置上，无地理限制。
- 3. DNS的主从结构非常稳定
- 4. 可以有效的分散DDOS攻击。
- 5. 你甚至可以在DNS服务商那里实现，自己不需要添加设备。而且没有带宽开销。

DNS负载均衡主要缺点

- 1. DNS负载均衡采用的是简单的轮循负载算法，不能够按照服务器节点的处理能力分配负载。
- 2. 不支持故障转移(failover)和自动恢复failback ,如果某台服务器拓机，DNS仍会将用户解析到这台故障服务器上，导致不能响应客户端。
- 3. 如果添加节点或撤出节点，不能即时更新到省市级DNS,可导致部分地区不能访问。
- 4. 占用大量静态IP。

1.1.2. 软件四层交换负载均衡

软件四层交换负载均衡为我们解决了几个问题

- 1. 能够按照服务器节点的处理能力分配负载。
- 2. 支持故障转移(failover)和自动恢复failback ,如果某节点拓机，调度器自动将它剔除，不响应客户端访问，当节点故障排除调度器立即恢复节点。
- 3. 可以随时添加节点或撤出节点，即时生效,方便网站扩容。

软件四层交换负载均衡优点

- 1. 仅仅需要一个静态IP。
- 2. 节点位于私有网络上与WAN隔离，用户面对的只是调度器。
- 3. 可以随时添加节点或撤出节点。
- 4. 通过端口可以组建多个集群。

1.1.3. 应用层负载均衡

Tomcat balancer

mod\_proxy\_balancer.so ,tomcat mod\_jk.so

MySQL proxy / MySQL-LB

1.2. 高可用性集群

俗称：双机热备份

关键词：心跳线

两部服务器，或多部服务器，形成一个集群，当主服务器崩溃是，立即切换到其它节点上。

两部服务器要做到，内容实时同步，保持数据一直。

一般用 heartbeat + DRBD 实现。heartbeat负责切换服务器，DRBD用于同步数据。

1.3. 负载均衡设备

负载均衡成熟产品

- 1. F5 Big IP
- 2. Array

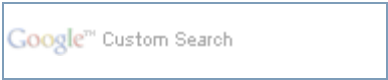
这些设备可提供3,4,7层负载均衡HA，硬件已经压缩，HTTP头改写，URL改写...

其中3层交换部分多采用硬件实现。

[更多关于F5与Array资料点击进入](#)

1.4. 会话保持

1.5. 健康状态检查



## 2. 缓存技术

首先要说明，很多缓存技术依赖静态化。下面展示了缓存可能出现的位置。

用户user -> 浏览器缓存 IE/Firefox Cache -> 逆向代理缓存 Reverse proxy Cache -> WEB服务器缓存 Apache cache -> 应用程序缓存 php cache -> 数据库缓存 database cache

当然交换机，网络适配器，硬盘上也有Cache 但这不是我们要讨论的范围。

缓存存储方式主要是内存和文件两种，后者是存于硬盘中。

网站上使用的缓存主要包括五种：

- 1. 浏览器 缓存
- 2. 逆向代理/CDN缓存
- 3. WEB服务器缓存
- 4. 应用程序缓存
- 5. 数据库缓存

将上面的缓存合理地，有选择性的使用可大大提高网站的访问能力。

总之，想让你的网站更快，更多并发，答案是cache,cache 再 cache

### 2.1. 浏览器缓存

#### 2.1.1. Expires

只要向浏览器输出过期时间HTTP协议头，不论是html还是动态脚本，都能被缓存。

HTML META

```
<meta http-equiv="Expires" content=" Mon, 10 Jan 2000 00:00:00 GMT"/>
<meta http-equiv="Cache-Control" content="max-age=300"/>
<meta http-equiv="Cache-Control" content="no-cache"/>
```

动态脚本

```
Expires: Mon, 10 Jan 2000 00:00:00 GMT
Cache-Control: max-age=300
```

```
Cache-Control: no-cache

header("Expires: " .gmdate ("D, d M Y H:i:s", time() + 3600 * 24 * 7). " GMT");
header("Cache-Control: max-age=300");
header("Cache-Control: no-cache");
```

很多web server都提供 Expires 模块

### 提示

有些浏览器可能不支持。

#### 2.1.2. If-Modified-Since / Last-Modified

If-Modified-Since 小于 Last-Modified 返回 200

```
neo@neo-OptiPlex-780:/tmp$ curl -I http://www.163.com/
HTTP/1.1 200 OK
Server: nginx
Content-Type: text/html; charset=GBK
Transfer-Encoding: chunked
Vary: Accept-Encoding
Expires: Mon, 16 May 2011 08:12:05 GMT
Cache-Control: max-age=80
Vary: User-Agent
Vary: Accept
Age: 38
X-Via: 1.1 ls100:8106 (Cdn Cache Server V2.0), 1.1 lydx156:8106 (Cdn Cache Server V2.0)
Connection: keep-alive
Date: Mon, 16 May 2011 08:11:23 GMT
```

If-Modified-Since 大于 Last-Modified 返回 304

```
neo@neo-OptiPlex-780:/tmp$ curl -H "If-Modified-Since: Fri, 12 May 2012 18:53:33 GMT" -I
http://www.163.com/
HTTP/1.0 304 Not Modified
Content-Type: text/html; charset=GBK
Cache-Control: max-age=80
Age: 41
X-Via: 1.0 ls119:80 (Cdn Cache Server V2.0), 1.0 lydx154:8106 (Cdn Cache Server V2.0)
Connection: keep-alive
Date: Mon, 16 May 2011 08:11:14 GMT
Expires: Mon, 16 May 2011 08:11:14 GMT
```

#### 2.1.3. ETag / If-None-Match

```
neo@neo-OptiPlex-780:/tmp$ curl -I http://images.example.com/test/test.html
HTTP/1.1 200 OK
Cache-Control: s-maxage=7200, max-age=900
Expires: Mon, 16 May 2011 09:48:45 GMT
Content-Type: text/html
Accept-Ranges: bytes
ETag: "1984705864"
Last-Modified: Mon, 16 May 2011 09:01:07 GMT
Content-Length: 22
Date: Mon, 16 May 2011 09:33:45 GMT
Server: lighttpd/1.4.26
```

```
neo@neo-OptiPlex-780:/tmp$ curl -H 'If-None-Match: "1984705864"' -I
http://images.example.com/test/test.html
HTTP/1.1 304 Not Modified
Cache-Control: s-maxage=7200, max-age=900
Expires: Mon, 16 May 2011 09:48:32 GMT
Content-Type: text/html
Accept-Ranges: bytes
ETag: "1984705864"
Last-Modified: Mon, 16 May 2011 09:01:07 GMT
Date: Mon, 16 May 2011 09:33:32 GMT
Server: lighttpd/1.4.26
```

### 2.2. CDN/逆向代理缓存

具有代表性的逆向代理服务器：

- 1. Squid



- 2. Nginx
- 3. Varnish
- 4. Apache cache module

其它逆向代理服务器

- 1. 一些提供cache的硬件设备
- 2. 最近几年出现了的 China Cache 服务商，也称CDN

很多CDN厂商使用Squid 二次开发做为CDN节点，通过全球负载均衡使用分发

这些CDN厂商主要做了一下二次开发

- 1. logs 日志集中
- 2. 流量限制
- 3. push,pull操作
- 4. url 刷新

s-maxage 与 max-age用法类似，s-maxage针对代理服务器缓存。同样适用于CDN

s-maxage 与 max-age 组合使用可以提高CDN性能

2.3. 负载均衡设备

F5 Big-IP, Array 等设备都提供硬件加速，其原理与squid, apache提供的功能大同小异

其中Array 页面压缩采用硬件压缩卡实现，SSL加速也采用硬件实现

2.4. WEB服务器缓存

例如，通过配置apache实现自身 cache

2.5. 应用程序缓存

在这个领域百花齐放，相信你一定能找到适合你的。这些cache会为你提供一些api，来访问它。

代表性的 memcached 据我所是sina广泛使用，腾讯也曾经使用过后来开发了TC(Tencent Cache)，台湾雅虎则使用APC Cache。

另外模板引擎也有自己的缓存系统

2.6. 数据库缓存

数据库本身就有这个配置选项，如果你仍然可以在数据库前面加一道Cache。

例如PostgreSQL, MySQL 都提供参数可以将memcached编译到它内部



### 3. 静态化

静态化方法包括：

- 1. 生成方式
- 2. 抓取方式
- 3. 伪静态化
- 4. 混合方式

静态化可以改善SEO

#### 3.1. 生成方式

主要由程序实现

例如

```
content = "<html><title>my static</title><body>hello world</body></html>"
file = open( your static file)
file.write(content)
file.close()
```

#### 3.2. 抓取方式

主要由程序实现

程序中抓取

```
content = get_url('http://netkiller.8800.org/index.php')
file = open( index.html)
file.write(content)
file.close()
```

使用软件抓取，不仅限于wget。

```
wget http://netkiller.8800.org/index.php -O index.html
```

这时只给出简单例子，使用复杂参数实现更复杂的拾取，然后将脚本加入crontab中可。

#### 3.3. 伪静态化

伪静态化是主要是通过URL上做一些手脚，使你看去是静态的，实质上它是动态脚本。

伪静态化实现主要包括两种方法：

- 1. Rewrite rule
- 2. path\_info

下面是一个PATH\_INFO例子

http://netkiller.8800.org/zh-cn/photography/browse/2009.html

根本就不存在这个目录'zh-cn/photography/browse/'和文件'2009.html'

下面是一个Rewrite例子

http://example.org/bbs/thread-1003872-1-1.html

### 3.4. 混合方式

其实目前网站使用的基本上都是上面几种方法混合方式。

例如首先将动态url(example.org/news.php?cid=1&id=1) 通过rewrite转换为 (example.org/new\_1\_1.html)

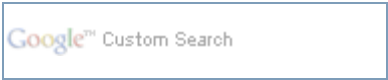
接下来就比较容易解决了，一种方法是使用wget example.org/new\_1\_1.html，另一种方法你无需静态化，直接使用squid规则配置让他永不过期

### 3.5. 静态化中的动态内容

在静态化页面中有一些内容是无法实现静态的。像登录信息，用户评论等等

我们用三种方法实现静态中嵌入动态内容：

- 1. iframe - 灵活性差
- 2. SSI - 消耗web服务器资源
- 3. Ajax - 依赖浏览器，稳定性差



## 4. 多媒体数据分离

### 4.1. 图片服务器分离

为什么要将图片服务器分离出来？

- 1. 图片通常比较大，下载需要更长的时间，而web容器并发数也是相当宝贵的仅次于数据库。
- 2. 传统浏览器一个窗口只占用一个链接数，目前主流浏览器都支持多线程下载，下载HTML页面同时，采用多线程下载其它多媒体数据。

我们举一个例子，你的服务器并发能力只用1000，早期浏览器不支持多线程，所以同一时刻，你的服务器可以承受1000个人同时访问。但现在不同了，基本所有的浏览器都支持多线程，假如你的页面中有9张小图片,同一时刻你的服务器仅仅能应付 $1000/10 = 100$ 个用户。

所以我们要将图片和其他多媒体文件分离出来，单独使用一台服务器处理请求。

#### 提示

图片服务器建议使用lighttpd与squid缓存配合使用效果更好或购买CDN的服务。

### 4.2. 目录层次规划

日期有利于归档

```
/www/images
/www/images/2008
/www/images/2008/01
/www/images/2008/01/01
```

分类不同用途的文件

```
/www/images
/www/images/theme/2009

# article id 000001
/www/images/article/2009/01/000001

# product id 00001
/www/images/product/2009/01/01/00001

# member name neo
/www/images/member/2009/01/01/neo
```

根据你的数据量，创建目录深度,并且目录深度有规律可循。

虽然64bit 文件系统不限制文件数量与目录深度，但是我还是建议按我的方式规划目录。

这样规划目录便于缓存控制，如：

images/2008/\* 永久缓存

images/2009/\* 缓存一个月

images/2010/\* 缓存一小时

images/2010/06/\* 缓存5分钟

### 4.3. 多域名访问

部分浏览器（IE）相同域名只能创建2个线程，在页面中使用多个域名可以解决这个限制

```
img1.example.com IN CNAME images.example.com.  
img2.example.com IN CNAME images.example.com.  
img3.example.com IN CNAME images.example.com.  
...  
imgN.example.com IN CNAME images.example.com.
```

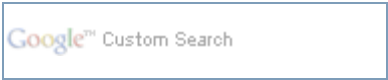


## 5. 压缩数据传输

服务器将html或脚本输出压缩，用户从服务器取得数据后由浏览器解压

压缩数据传输实现方法：

- 1. apache mod\_deflate
- 2. lighttpd compress module



## 6. 时间同步

将通信网上各种通信设备或计算机设备的时间信息(年月日时分秒)基于UTC（协调世界时）时间偏差限定在足够小的范围内（如100ms），这种同步过程叫做时间同步。

关于时间同步我个人的解决方法：

- 1. 使用UTC时间，用户加时区来解决。
- 2. 保证所有服务器的时间是同步的

```
$ sudo ntpdate asia.pool.ntp.org
21 May 10:34:18 ntpdate[6687]: adjust time server 203.185.69.60 offset 0.031079 sec
```





## 7. 邮件系统

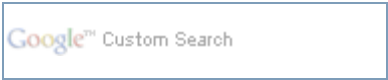
邮件系统：

- 1. 站内邮件。
- 2. 电子邮局服务
- 3. 订阅/推广邮件

### 7.1. Mailing List

邮件列表系统：

- 1. 订阅功能
- 2. 确认订阅功能
- 3. 退订功能
- 4. 群发功能
- 5. 浏览功能(国内基本不需要)



## 8. 日志集中管理

传统做法是文件型日志，分布在各个服务器上。在大规模部署服务器后代来很多不便，增加很多管理成本，所有我们需要集中管理服务器产生的所有日志，我们叫他日志中心服务器

### 8.1. 系统日志

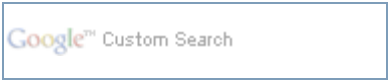
`rsyslog/syslog-ng` 实现日志集中管理

### 8.2. 应用程序日志

应用程序中没有比较大量记录日志，当开启debug模式时才记录大量日志。

但是很多国内开发太过于依赖日志，导致日记非常臃肿

程序日志解决方案，请看软件架构相关章节



## 9. SSL

SSL 加密传输，为电子商务提供交易安全保护，什么时候该使用 SSL呢：

使用SSL

1. 用户登录
2. 购物流程
3. 支付

什么时候不使用SSL? 经过SSL加密后，你就失去了很多功能，你不能在对页面做Cache/CDN，SSL加密与解密需要耗费你的服务器CPU与内存资源，能不使用尽量不使用。

对于SSL消耗你服务器资源这方面有两个方案解决

1. 将SSL证书安装到CDN上，目前蓝讯，网宿等等CDN厂商都提供SSL服务。我与上两家技术人员沟通过，也安装了证书实际测试一下，你可以放心是使用。
2. 将SSL证书安装到负载均衡设备，这些设备都采用专用硬件处理SSL请求，我测试过F5，Array，Banggoo

采用上面两种方案，无需改变你目前的服务器配置，他们的原理是

user (https://www.example.com) --> CDN or SLB (SSL) --> http://www.example.com

用户访问https,到达CDN或者负载均衡，CDN/SLB 通过http://请求源站，然后将内容SSL加密，返回给用户，这样用户得到的是加密内容。

用户提交数据，交给CDN/SLB，CDN/SLB将SSL加密数据卸载证书，然后将解密后数据发回源站。

CDN与SLB加载卸载证书原理很简单，不难理解。

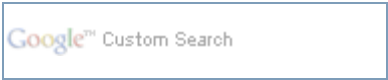
我来教你DIY一个，你可以使用Squid，Nginx，Apache等等反向代理服务器，将证书安装在反向代理上，请求源站仍然采用http。

### SSL注意事项

你如果认为把SSL挂载到网站前端就，大功告成，完事了，那你错了。

幸运的话你会成功，但有的时候你发现你的证书不被信任。如果你是个细心的人，你会发现单个图片，或者你创建换成测试文件 `echo helloworld > index.html` 证书都是OK的。

这个问题出在你的html页面中，安装有SSL证书的网站，不能有外链js,flash等等不安全内容。



# 10. Storage 存储

## 10.1. 存储种类

DAS、NAS、SAN

### 10.1.1. Direct Attached Storage

PC + Raid Card ===== Array

### 10.1.2. Network-attached storage

NAS 说白了就是一个嵌入式电脑，经过精简内核的Linux,通过samba,nfs,WebDav,ftp...等等方式实现共享存储

如果你有兴趣，可以DIY一个NAS，使用Openfiler

### 10.1.3. Storage area network

只要有¥什么都好说

#### 10.1.3.1. FC SAN

FC 是光纤通道网络存储，需要专用交换机与HBA卡

提供 6G/8G 数据传输

#### 10.1.3.2. IP SAN

1G/10G iSCSI，采用TCP/IP协议传输SCSI指令

客户端不需要专门的HBA卡，专业iSCSI HBA目前非常昂贵

#### 10.1.3.3. FCoE (Fibre Channel over Ethernet)

因为iSCSI很廉价，FC市场被iSCSI蚕食，传统FC收到iSCSI压力。推出新一代协议，希望能在现有光纤通道的成功基础上，借助于以太网的力量重新保持自身在数据中心存储局域网中的霸主地位。

iSCSI通过TCP/IP协议在可能产生损耗或阻塞的局域网和宽带网上传送数据存储块。相比之下，FCoE则只是利用了以太网的拓展性，并保留了光纤通道在高可靠性和高效率方面的优势。

## 10.2. RAID

### 10.2.1. 缓存服务器

全部采用RAID 0

一旦出现问题，立即将其从集群中踢出去，带节点故障排除后，恢复它的功能。

### 10.2.2. Web 服务器

采用RAID 1

服务器仅仅存放脚本程序，数据建议放在外挂存储上。

### 10.2.3. 数据库

主服务器：建议采用 RAID 10

数据库节点：建议采用 RAID 10

数据库应尽量避免使用RAID 5，RAID 5在做校验过程时，效率会很低。

数据库节点一旦出现问题，立即从集群中撤出，排除故障后，在回复使用。

### 10.2.4. 数据备份

数据备份服务器建议采用RAID 5/6

RAID 5 阵列容量计算公式：

可用容量 = (n-1) /n的总磁盘容量 (n为磁盘数)

## 10.3. File System 文件系统

我个人推荐使用 ext4, xfs 或 reiserfs

zfs 也不错

### 10.3.1. Distributed File System(DFS)

RAID 0提高吞吐能力是有限的，IO也会有瓶颈，NAS吞吐能力一样有限，SAN价格不菲。

DFS是一个不错的选择

## 10.4. 数据访问协议

- 光纤通道管理
- iSCSI
- IP/RDMA
- iSER

- SRP
- NFS v3 和v4
- CIFS
- HTTP
- WebDAV
- FTP
- NDMP v4

10.5. 数据管理

10.5.1. Share 共享

10.5.2. Mirror 远程镜像同步

10.5.3. 压缩与重复数据消除

EMC Data Domain

开源 Openedup

10.5.4. Backup 备份与恢复

Bacula/Zmanda

10.5.5. 故障报告

[Home](#) | [Mirror](#) | [Search](#)



## 第 3 章 Software architecture (软件架构)

*Software architecture*

目录

[1. 开发框架 Framework](#)

- [1.1. HMVC](#)
- [1.2. REST](#)
- [1.3. SNA \(Shared Nothing Architecture\)](#)
- [1.4. 其他](#)

[2. URL设计](#)

- [2.1. URL 作为MVC 的Controller](#)
- [2.2. URL 伪静态化，用于SEO优化](#)

[3. Javascript Framework](#)

- [3.1. Minify JS](#)
  - [3.1.1. jsmin](#)
  - [3.1.2. yuicompressor](#)
  - [3.1.3. shrinksafe](#)

[4. 模板\(template\)](#)

- [4.1. HTML 页面优化](#)

[5. Session/Cookie](#)

- [5.1. Session](#)
- [5.2. Session 共享](#)
- [5.3. Cookie](#)
- [5.4. Cookie 安全](#)

[6. 国际化 Locale database。](#)



[6.1. Unicode](#)

[6.2. 繁简转换](#)

[7. 数据库访问](#)

[7.1. CRUD](#)

[7.2. Active Record](#)

[7.3. OR Mapping](#)

[8. Cache](#)

[8.1. 页面缓存](#)

[8.2. 局部缓存](#)

[9. Single sign-on \(SSO\) 单点登录](#)

[10. 搜索引擎](#)

[11. Synchronous/Asynchronous](#)

[12. Message Queuing](#)

[13. Hash](#)

[14. Sharding 垂直/水平切割](#)

[14.1. 面向服务](#)

[14.2. 面向数据库](#)

[15. 日志系统](#)

# 1. 开发框架 Framework

选择一个好的开发框架，很重要。不过大部分框架都针对于软件开发，而我们要的是轻量级，适合高负载，灵活的框架。

框架的分类

- HMVC
- REST

上面框架可以满足我们绝大多数需求，如URL定义，Session/Cookie管理，多语言国际化，数据库访问等等。

Java和.Net我没有太多的经验，php我有10+年经验，我在各种框架之间做比较发现CodeIgniter框架比较适合我们的需求。

框架是没有100%完美的，你仍需要对它进行二次开发。如果你有充足的时间，针对自身系统系统的特点设计一个更适合您网站的框架，这是最好的选择。

设计一个框架需要用到很多知识，需要有丰富的经验。目前主流框架都是基于MVC设计思想，要设计一个框架

你必须了解MVC (Model-View-Controller) 参考：<http://www.itisedu.com/phrase/200604231324325.html>

开发一个框架包括那些重点呢，下面我把一些要点一一列出，然后一个个地突破，我这里使用php为例子，上面我已经说过语言只是工具，所以学习是设计思想，不要拘泥于语言：

- JS封装（javascript）
- 模板（template）
- url
- session/cookie
- 语言包（language package）
- 编码（unicode）
- 数据库访问（database OR Mapping）
- 权限（Permission）

如果重新开发一个框架，我认为太现实，我的建议使用现有pear库，搭建一个MVC框架。例如：

- Model（pear db）
- View（smarty template）
- Controller（pathinfo）

1.1. HMVC

- Python web2py
- Php CakePHP,Zend,CodeIgniter
- Perl Catalyst
- Java Struts, Spring MVC
- ruby on rails

1.2. REST

- Python Pylons
- ruby on rails

1.3. SNA (Shared Nothing Architecture)

1.4. 其他

- Python Django 是一个MTV框架
- .Net Framework



## 2. URL设计

一个大型网站，对于URL规划我认为非常重要，这也是为什么我把它单列出来的原因。

当前网站上使用的URL虚虚实实已经不单单是划分目录空间功能，它与程序配合使用，实现复杂的逻辑功能。在应用程序开发框架组成中占有重要的地位。

### 注意

无论什么文件系统，每个目录下容纳的子目录和文件是有限制的，并且内容过多会影响文件索引速度，所以合理地划分目录空间很重要

下面是URL实例仅供参考，稍后我会详细解释他们这样设计的目的是什么和实现方法。

- `http://sina.allyes.com/main/adfclick?db=sina&bid=120294,154641,159584&cid=0,0,0&sid=146767&advid=2618&camid=19961&show=ignore&url=http://web.topxue.com/gj/bdxm/`
- `http://news.sina.com.cn/c/2008-05-22/172315597145.shtml`
- `http://example.org/bbs/thread-1003872-1-1.html`
- `http://example.org/news/2008/05/22/1004862.shtml`
- `http://example.org/uk/en/action,ProductDetailShow_productId,51`
- `http://example.com/forums/viewforum/59/`
- `http://example.com/forums/viewthread/80165/`
- `http://trac.example.com/cgi-bin/trac.cgi/ticket/1286`

目录设计，以下为真实目录，你在URL看到其它路径都是不存在的。它们是用于rewrite或pathinfo的。

- `images`
- `framework`
- `model`
- `view`
- `controller`
- `language`
- `config`
- `logs`

### 2.1. URL 作为MVC 的Controller

例子1

http://example.com/guestbook/view/59/

相当于

http://example.com/<controller>/<action>/<id>/

```
class Guestbook extend Controller{
    public function index(){
    }
    public function view($id =1){
    }
    public function add(){
    }
    public function remove($id){
    }
}
```

一般采用pathinfo技术实现上述功能

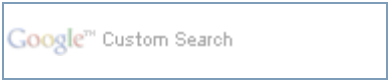
2.2. URL 伪静态化，用于SEO优化

http://example.com/guestbook/view/59.html

相当于

http://example.com/guestbook.php?action=view&id=59

一般使用Rewrite技术实现



### 3. Javascript Framework

javascript是面向过程的，只要请引用.js文件即可访问他的方法（function），并且传统方式会定义很多全局变量。如果大量使用javascript难免会出现变量覆盖，或function同名。

所以我们要将javascript封装成class，另一点我们也需要面向对象支持。下面是几个常用的JS开发框架（javascript framework）。

- prototype
- jQuery
- mootools
- script.aculo.us
- Dojo
- MochiKit
- rico

#### Javascript GUI

- ExtJS
- qooxdoo

#### 3.1. Minify JS

##### 最小化js文件

###### 3.1.1. jsmin

http://crockford.com/javascript/jsmin

```
jsmin <fulljslint.js >jslint.js
```

###### 3.1.2. yuicompressor

http://developer.yahoo.com/yui/compressor/

Usage: java -jar yuicompressor-x.y.z.jar [options] [input file]

3.1.3. shrinksafe

http://dojotoolkit.org/docs/shrinksafe



## 4. 模板(template)

模板最早是在cgi程序中广泛应用，cgi是动态页面的第一代，同期还有NSAPI,ISAPI,第二代是fastcgi,asp,php,ColdFusion...第三代是.net与java。

模板的特点：

- 模板可以分离代码和页面
- 模板能够改善页面结构
- 模板可实现页面重用
- 模板可以区块化，如同搭积木
- 设计人员不需要关心代码
- 实现主题

模板有很多优点，但它也会增加系统开销，不过我们可以通过cache来解决这个问题。

常用模板引擎：

- PHP: smarty template
- Perl: TT template
- Python： Cheetah

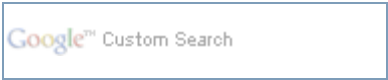
### 4.1. HTML 页面优化

页面减肥

```
{strip}
<html>
....<head>
.....<title>Title</title>
....</head>
....<body>
.....<h1>Hello world</h1>
.....<div class="">
.....Test
.....</div>
....</body>
</html>
{/strip}
```



Smarty 的{strip}可以删除页面中的空格，Tab符以及回车换行符



## 5. Session/Cookie

为什么我要在这里提Session和Cookie，这也大型站点必须要处理问题。

### 5.1. Session

在集群环境中与单服务器是不一样的，集群组成可分为调度服务器和节点，节点数量不定，单个节点安装有web服务器，用户每次访问网站调度服务器随机分配一个节点给该用户， 举一个例子：用户在网站上看新闻，点击第一个连接被分配到node 1上去，当他看完这条新闻并单击下一条时，可能被分配到其它节点上，这里刚才建立的session在node 1上， 它就会因失去session而必须重新登录。

所以我们要同步所有节点上的Session, 另外如果能用Cookie代替Session的地方尽量使用Cookie。

### 5.2. Session 共享

解决方案：

1. 不用Session，使用Cookie取而代之
2. 共享Session，放到数据库中，放到Memcache中

PHP Session很有解决方案：

查看PHP手册 Session Extensions 章节，重写Session逻辑。

共享Session用Memcache，在php.ini中配置即可

```
session.save_handler = memcache
session.save_path = tcp://127.0.0.1:10001
```

### 5.3. Cookie

Cookie 我这里提到cookie是可以实现“单点登录”功能。

一个网站可能不指一组集群系统，如news.example.org, bbs.example.org, blog.example.org 要实现在一处登录即可在其它站点上同时也处于登录状态，就要用到Cookie来实现。

### 5.4. Cookie 安全

Cookie存储在用户端，Cookie数据极易伪造。下面提供几个方案。

- 在Cookie数据上加干扰词
- 在反向代理上做手脚
- 负载均衡设备都提供Cookie保护功能



## 6. 国际化 Locale database。

在开始具体介绍之前，需要先介绍几个术语：

- i18n: 就是internationalization, 国际化,由于首字母"i"和末尾字母"n"间有18个字符，所以简称i18n.  
internationalization指为了使应用程序能适应不同的语言和地区间的变化而不作系统性的变化所采取的设计措施。
- l10n: 就是localization, 本地化，由于首字母"l"和末尾字母"n"间有10个字母，所以简称l10n. localization指为了使应用软件能够在某一特定语言环境或地区使用而加入本地特殊化部件和翻译后文本的过程。
- locale: 简单来说是指语言和区域进行特殊组合的一个标志，如：en-us, zh-cn, zh-tw

l10n有很多历史遗留问题，l10n目前已经被i18n取代。

我自己曾经使用过下面四种方式实现语言包

1. 定义一个数组
2. 使用数据库
3. 使用文件
4. 使用数据结构

### 6.1. Unicode

相比几年前，目前各种语言对UTF-8支持都比较好。

在BBS上常常看到一些网友抱怨UTF-8出现“乱码”问题，让我们看看都有哪些地方涉及编码问题。

用户输入法->IDE开发环境,浏览器->web容器->数据库

任何一个环节出现问题有可能出现问题

- 首先是输入法，早期输入法可能是GB2312或GBK。
- 其次是IDE开发环境，当你创建一个空文件时，它的已经具备某种编码，一般外国开发工具默认是acsii，这一点我认为Dreamware做的最好，可以随时切换编码。
- 浏览器现在基本不用担心

- web容器apache 2.x对unicode支持很好，tomcat本身机器码就是unicode。
- 数据库问题也不大，PostgreSQL相比MySQL对Unicode支持也早，也比较好。MySQL这方面有点复杂。

提示

如果你不考虑使用Unicode并且想支持繁体和简体中文，你可以使用GBK，但我建议你使用GB18030。

Unicode不是最好的选择，它占用三个字节，数据量较大，选择适合你的编码，如果你是英文网站，请使用ISO-8859-1，如果是简体中文，请使用GB2312

6.2. 繁简转换

方案1: User -> Squid ->Web Server

修改squid源码，加入iconv(big5,gb2312,html page)

```
e.g.1 user (gb.example.org) -> Squid (big5->gb2312) -> web server
e.g.2 user (big5.example.org) -> Squid (gb2312->big5) -> web server
```



## 7. 数据库访问

早期php访问数据库的做法是写一个连接文件，include包含进来，然后在页面使用sql操作函数，返回结果。

- CRUD (create, read, update and delete)
- Active Record
- OR Mapping

另外设计一个框架是还要考虑，切割表，分库。

### 7.1. CRUD

### 7.2. Active Record

### 7.3. OR Mapping



## 8. Cache

Cache大体分为两种，一种是文件Cache,另一种是内存Cache。按应用划分，可以分为页面Cache和局部Cache

### 8.1. 页面缓存

页面缓存有三种实现方式：

1. 反向代理
2. HTTP Header
3. Template 层

页面cache的原理是推送HTTP协议头，修改网页过期时间。

### 8.2. 局部缓存

局部cache是可以将数组，序列化对象，字符串等等，分别cache，并设置ttl值（生存时间）

数据库与应用程序之间加一层Cache,性能将会大幅提升。

我们一般会把Cache封装为一个Class (类)，并且支持多种Cache API.如： Apc Cache,Xcache, Memcache...

Cache操作很简单，添加，更新，删除，状态几种操作，其中添加与更新可以共用一个方法。



## 9. Single sign-on (SSO) 单点登录

提供一站式登录，即一旦在\*.domain.com任何位置登录后， 全域均不需要在重新登录

解决方案：

- 1. 登录 cookie 的host 使用domain.com 不要使用www.docmain.com
- 2. 子域名通过rewrite实现,即bbs.domain.com = www.domain.com/bbs





# 10. 搜索引擎



# 11. Synchronous/Asynchronous

举几个例子

synchronous (同步)

- 1. 用户登录
- 2. 商品价格与相关计算
- 3.

asynchronous(异步)

- 1. 批量操作建议使用异步方式
- 2. 消息广播操作
- 3.

用户注册，发帖与评论。采用同步或异步均可，还要看你的具体情况而且。



## 12. Message Queuing

Message Queuing 有 synchronous (同步)/asynchronous(异步) 之分，不同场景适合不同的处理方式。

[RabbitMQ](#)

[ZeroMQ](#)

[Apache ActiveMQ](#)



# 13. Hash



# 14. Sharding 垂直/水平切割

## 14.1. 面向服务

## 14.2. 面向数据库



# 15. 日志系统

开源的日志系统，包括facebook的scribe，apache的chukwa，linkedin的kafka和cloudera的flume等

Scribe: <https://github.com/facebook/scribe>  
Chukwa: <http://incubator.apache.org/chukwa/>  
Kafka: <http://sna-projects.com/kafka/>  
Flume: <https://github.com/cloudera/flume/>



# 部分 II. Development

目录

[4. 开源模式](#)

[5. 开发语言及平台](#)

[1. 分层架构](#)

[1.1. 中间件 Middleware](#)

[1.2. 分层](#)

[2. Web 2.0](#)

[3. 云计算](#)

[4. 跨平台](#)

[5. 编译语言比脚本语言安全](#)

[6. 封装重用](#)

[7. 相关的工具](#)

[7.1. 开发工具](#)

[6. project](#)

[1. Project directory](#)

[7. Logging](#)

[1. Remote Syslog](#)

[8. RDBMS 数据库设计](#)

[1. 数据字典](#)

[2. 用户帐号表](#)

[3. 分类表设计](#)

[3.1. 树形分类表](#)

[3.2. 多对多分类](#)

[3.3. 快速检索子分类设计](#)

[4. 文章表设计](#)

[4.1. 分区表设计](#)

[4.2. Title性能优化](#)

[5. 评论表](#)

[6. 记录点击率，阅读次数，及评分表](#)

[7. 产品属性表](#)

[7.1. 简单实现](#)

[7.2. 实现属性组管理](#)

[7.3. 可编辑属表](#)

[8. 国际化语言表](#)

[9. Workflow](#)

[10. 内容版本控制](#)

[11. Sharding](#)

[11.1. horizontal](#)

[11.2. vertical](#)

[9. 页面设计 Design](#)

[1. XHTML+DIV+CSS](#)

[2. 页面结构设计](#)

[2.1. Home page \(首页\)](#)

[2.2. 导航栏](#)

[2.3. Left Bar](#)

[2.4. 区块设计 Block](#)

[3. 表格](#)

[4. 图片优化](#)

[4.1. onMouseOver/onMouseOut](#)

[4.2. 使用一幅图片处理BLOCK四角](#)

[4.3. 图片用背景图代替img 标记](#)

[5. HTML嵌入图片](#)

[6. 页面内容安全](#)

[6.1. 禁止鼠标右键](#)

[6.2. 禁止复制剪切 及粘贴](#)



[7.html,css 有效性检查 Validation](#)

[10. Framework Design](#)

[1. MVC Framework](#)

[2. HMVC Framework](#)

[3. REST](#)

[3.1. RESTful JSON API](#)

[4. action 分发器](#)

[5. Plugin & Hook 设计与实现](#)

[5.1. 插件管理平台](#)

[5.2. 接口定义](#)

[5.3. 插件](#)

[5.4. 测试](#)

[6. Cache](#)

[6.1. CDN \(Content Delivery Network\) 与反向代理](#)

[6.1.1. CDN接口API](#)

[6.1.2. 方向代理页面过期处理](#)

[6.1.3. 内容版本化](#)

[6.2. HTTP Cache](#)

[6.2.1. 在程序中灵活操作 Cache-Control](#)

[6.2.2. 非程序文件缓存处理](#)

[6.3. Cache 生存时间](#)

[7. Session/Cookie 安全](#)

[7.1. cookie-free domains](#)

[8. i18n 国际化](#)

[8.1. 数组方式](#)

[8.2. 数据库方式](#)

[8.3. 文件文件](#)

[8.4. Gettext](#)

[8.5. 数据结构](#)

[9. Logging](#)

---

10. SEO Module

[10.1. robots.txt](#)

[10.1.1. Sitemap in robots.txt](#)

[10.2. sitemaps](#)

[13. Application Design](#)

[1. Product](#)

[2. Cart & Checkout](#)

[3. Other](#)



## 第 4 章 开源模式

我在IT行业干了8年，做过大大小小的公司不少，项目管理上有乱来的，有ISO国际化的，先进的CMMI过程的，还有开源方式的。我比较趋向开源模式，最近几年一直在外企背景的本地公司，开发模式采用开源模式的企业应用模式。近年来开源模式有颠覆传统商业模式趋势，很多公司开始寻求开源盈利模式，IBM，Oracle是非常成功的，Sun反映不及时，受到冲击最大。具有代表行的开源盈利模式是MySQL, Redhat。

开源模式，没有那么多条条款款限制，比较灵活，反应速度快。并且适合任何规模的项目，小到几个人，大到上千人。其特点为拥有3-5名核心维护人员，参与开发的人员10人-40人之间，采用SVN进行代码管理，通过maillist/irc进行开发交流，有明确的开发计划和日程。

开源没有严格等级的组织架构，团队领导仅仅是组织/协调工作，合并代码。开发人员比较分散，可能两地，三地，甚至更多参与开发。并且同时进行开发，多个模块向前推进。来完成一项伟大的工程。成员有什么好想法，就发布在mailing list上，大家讨论，确认下来，你就可以开始开发。如果与大家不同意你的idea，你可以产生一个项目分支。

这种模式对参与人员能力要求比较高，要求能独立完成任务，有创意，自觉性强，团队合作意识强，。

开源模式也有它的缺点，不能一概照搬，如果照搬开源模式，显而易见人力成本太高了。因为开源成员都是精英及大师黑客，一个开源项目团队就像海军陆战，单兵作战以一抵十，团队作战所向披靡。而且黑客的个性很强，企业不一般不需要员工有个性和创造力，这不便于管理。只要按需求做，不出错误就是好员工。

目前国内企业仍是以高级工程师为核心带领年轻的程序员或应届毕业生方式进行项目开发。

我不得不说中国人很爱跟风，接受能力最快。什么技术流行我们就用什么，最新的技术应用都在中国，你会发现我们的技术是最先进。更本不等市场验证。

在国内企业中你会发现很多眼花缭乱的术语，技术及软件全部用在项目中，向 UML, Visio, Project, Rose, FreeMind, ER-Win, ISO, CMM, GB...

你会发现我们太先进了，Project管理项目，Subversion/VSS/ClearCase控制版本，UML建模，我们用ER-Win/Power Design设计数据库，我们文档用CMM格式，MVC开发框架，ORM操作数据库，我们用Load Runner测试，我们用QQ/MSN沟通。我们在按着别人的思想，别人定义的标准，帮别人验证他们想法是正确的。如果失败了，就推到重来。

我们的外国客户，还在用ssh登录vim/emacs开发，还再用CVS还是命令行的，Trac/Wiki管理开发资源，开发文档就是一个简单功能说明，在邮件列表上沟通





# 第 5 章 开发语言及平台

## 目录

### [1. 分层架构](#)

#### [1.1. 中间件 Middleware](#)

#### [1.2. 分层](#)

### [2. Web 2.0](#)

### [3. 云计算](#)

### [4. 跨平台](#)

### [5. 编译语言比脚本语言安全](#)

### [6. 封装重用](#)

### [7. 相关的工具](#)

#### [7.1. 开发工具](#)

语言只是一个工具，一种实现我们需要的工具，每种每言都有它的优点和缺点和，在不同领域发挥各自的长处，并且都有它存在的意义。

语言不段地发展，市场决定它们是生存还是没落走向死亡。只有最活越，生命力强的语言才能生存下来。.net与java后面是强大的财团做后盾，大量被捆绑的客户支持他，并有完备的客服，从商业角度选择它是没有错误的。

但不要拘泥与语言，使用你最熟悉的语言,选择你最擅长的数据库和操作系统。

目前大型网站都不会单一选择一种语言和数据库，一种操作系统。例如:

- 淘宝，前端展示页面采用php,后台管理采用java...
- Yahoo，主要使用php开发，但搜索引擎采用java...
- Ticketmaster，前端展示采用perl开发，后台管理java, 前台展示数据库采用mysql,会员数据库采用oracle

像php/perl/python这种动态语言，开发速度快，周期端，对服务器性能要求低，出错率低，他们的设计这希望它能尽量使代码运行下去， 而不是抛出异常，终止执行或崩溃。而行对于开发者要求门槛比较低。 php 无论怎么开发都不会使web server 崩溃。而 Java 则不同，很容易崩溃。

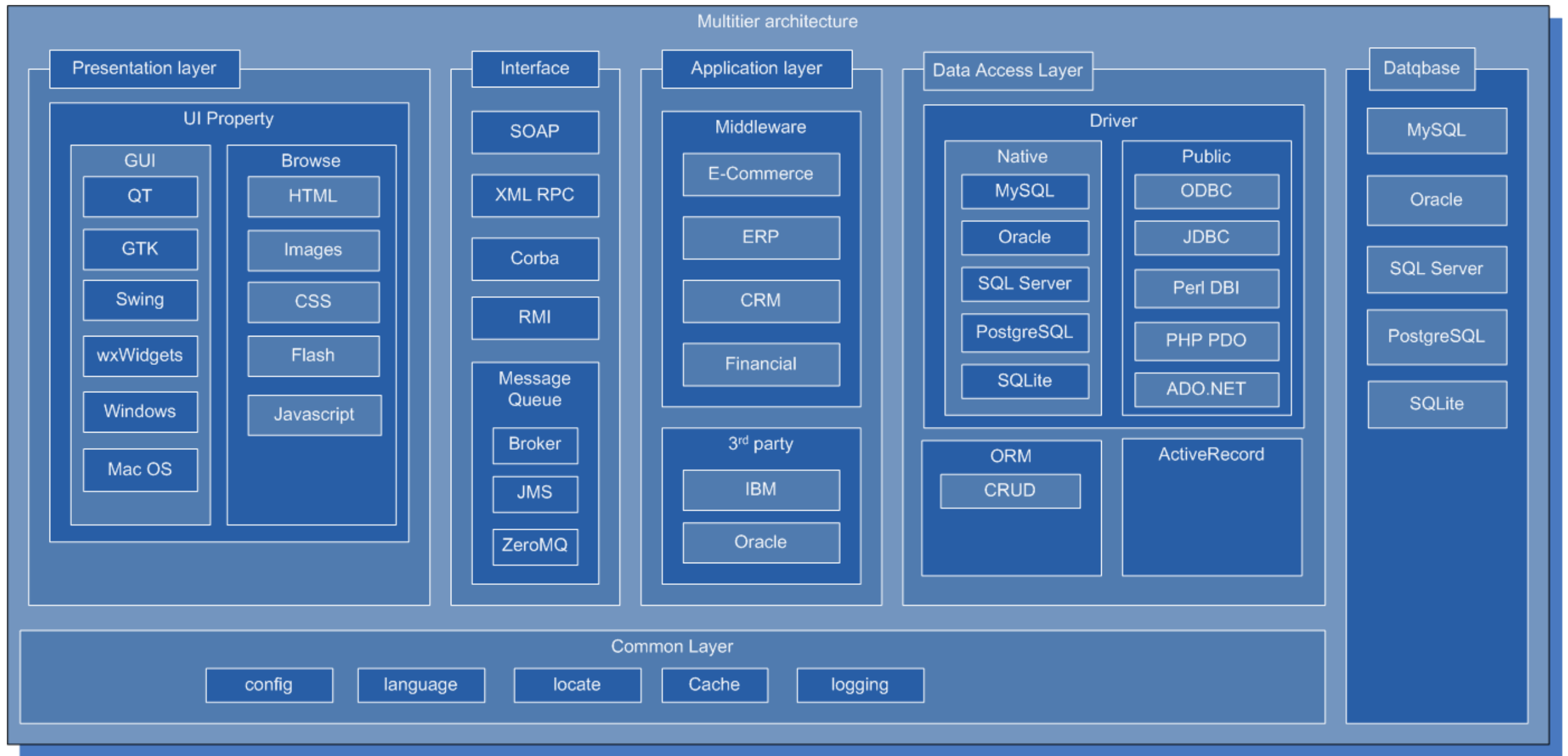
Java 我认为java是个非常不错的语言，错在JVM上。这是一个垃圾的解释器，效率极差。不加优化的，把所有东东全部load进内存。采用java技术，开发成本相当高，对开发人员要求很高，而且需要一个稳定的团队。国内资深java开发人员大多转向管理层。 只有细心的人才能驾驭Java，否则不能保证软件质量，我在工作中发现php团队开发的代码质量明显比java高，bug 数量上比Java代码的 bug 少很多。 我个人认为少于5年工作经验程序员很难写出一流的Java程序。 Java 架构最不能容忍的是有时不得不restart才能生效。而写的很烂程序你不得不采用restart来保证系统正常。

.net 不是很熟悉，.net 开发环境最好，速度比java快，只要有钱，全用正版，选择微软的产品很不错。很多linux爱好者鄙视M\$, 对windows系统很有很大偏见，偏激。 window系统很稳定，并非像网上传的那样不堪一击，很多引起windows崩溃的原因是硬件问题。我在工作中发现国产服务器在板卡接口上做工不过关。 没有镀金或防氧化处理，导致内存丢失，cpu丢失，RAID丢失... 等等引起系统崩溃。但linux系统确能运行下去，不过一旦重启，将不能恢复。

## 1. 分层架构

### 1.1. 中间件 Middleware

<http://en.wikipedia.org/wiki/Middleware>



很多人谈到java就会涉及到三层架构即：web 容器 -> application server 应用服务器，中间件 -> 数据库

三层架构其实不是什么新鲜东西，J2EE仅仅是对象请求代理体系结构的一种，任何语言都能实现三层架构。中间件不是Java专利

- 编译执行的语言基本都支持Corba
- python 则有Zope，Zope是一个很成功App Server。足以比肩J2EE. python 也支持corba库，我尝试过python -> corba -> PostgreSQL.但性能不佳。

php/perl 一样可做到,采用SOAP, XML-RPC等技术,可以实现部分功能。但我们可以架构上做些改变。

总之，不要拘泥于三层架构，仅仅是实现方式的一种。没有最合理，也没有最好的，根据你的需求作出调整，最终是看结果，而不是实现过程。

1.2. 分层

中间件的概念

```
web ----> /--> app server ---\
           ----> app server -----> Database
           \--> app server ---/
```

php 分层

```
load balance ----> /--> web server ---\
                   ----> web server -----> -> Database
                   \--> web server ---/

load balance ----> /--> web server ---\ /--> app server ---\
                   ----> web server -----> -- SOAP/XMLRPC--> ----> app server -----> Database
                   \--> web server ---/ \--> app server ---/
```

你可以采用复核型架构，我曾经就做过这样的项目php -> soap -> .net framework

```
apache ----> php /--> module ---\
                ----> module -----> MySQL Database
                \--> module ----> soap ----> IIS ----> DCOM ----> C# script ----> SQL Server
                \-> module ----> tomcat ----> jasper report ----> Oracle
```

我们系统有一个需求是，在php中创建word,excel文档，编辑文档，比较文档..., 我们需要调用office.dll实现

同时我们有一个报表系统，是通过jasper report 实现的

另外fastcgi

```
web server ----> /--> fastcgi process ---. \
                  ----> fastcgi process -----> Database
                  \--> fastcgi process-----' /
```







## 2. Web 2.0

商业炒作产物，对于纯高技术的人来说，虚头



### 3. 云计算

云计算还是个概念，但在云计算大潮中，不支持云计算，显得没有技术含量。

被网络炒得“神乎其神”。起初我误以为是分布式计算的下一代，后来发现和分布式计算根本两个不同的东西。与网格计算/分布式计算扯不上边。

"云计算"这个词已经被泛滥使用，

比较靠谱是亚马逊EC2 其实就是一堆Xen虚拟机, Dell说他的刀片服务器是云计算，Vmware 也说是云计算，Oracle 说他的VirtualBox是云计算，说ZFS是云存储。我也说不清楚，自己斟酌。



## 4. 跨平台

没有真正的跨平台语言，所谓跨平台都是忽悠人。

只要提供不同平台的编译器加条件编译，即可实现跨平台。或提供不同平台的解释器，也可一实现跨平台。

例如大家都很看中Java的跨平台，但想一下，这个“跨平台”是要打引号的，实际上这个跨平台准确的说是跨Sun提供的标准JVM平台，而非OS平台。只要某个JVM支持某个OS，你的程序才可以跨过去。如果JVM不支持这个OS平台，Sorry，你的程序不可能跨过去。不信你去java.sun.com下载jre你会发现仅提供四个平台版本Linux,Mac OS X,Solaris,Windows

很多OS都不支持Java。如FreeBSD 就不支持Java,必须使用Linux glib 运行Java，效率很低,IBM用的是IBM 自己开发JVM 至于他和Sun Java有什么关系，可能是授权。

只要能让你的程序翻译成JVM字节码，你的程序就可运行在JVM上。如：

php通过Quercus(<http://quercus.caucho.com/>)把PHP文件编译成.java文件,让后javac编译成class文件后在一些JavaEE应用容器中运行PHP程序

Jython可以将Python编译成java文件

JRuby可以将Ruby编译成java文件

虚拟机并非只有JVM，还有Parrot, Perl6 就是在Parrot虚拟机上实现的。

另外开发一种新语言也并非难事，只要你有时间精力投入我想不出3年，就可以打造一门新语言。

开发新语言也并非难事，只要你有时间精力投入我想不出3年，就可以打造一门新语言。



## 5. 编译语言比脚本语言安全

错！

编译不能保证代码安全，仅仅能保证你的代码不被人使用。

但目前中国人力成本相当的便宜，重新实现你的功能逻辑并非难事。所以只要你的网站上线，在很短的时间内就可以出现很多山寨版。

编译流行的原因是为了解决微机的速度以及存储问题，随着微机处理器技术突破，你根本不用担心速度问题。中型机与大型机领域脚本语言站多数。



## 6. 封装重用

重用可以减轻劳动，但过分重用，会牵一发动全身。

尤其对于二次开发者不熟悉你的系统，导致修改一个bug，又产生新的bug。

另外模板也不宜拆分的过于零碎。模本的组装，需要很多时间并很消耗你的资源。



## 7. 相关的工具

### 7.1. 开发工具

#### Mozilla Firefox 及扩展

- Web Developer
- Firebug 调试必备工具
- YSlow 性能分析工具
- Live HTTP Headers 相当于HTTP Sniffer嗅探器，可以跟踪HTTP协议头，调试cache时比较有用。
- IE Tab 用于IE/Firefox之间切换
- FoxClocks 如果开发工作跨时区，这个很有用
- Foxmarks Bookmark Synchronizer/Weave 将开发资源放入书签，同时在开发团队中保持同步
- Fasterfox 可以一显示页面载入时间，方便页面优化. 如果安装了YSlow可以不装这个插件。
- FireFTP
- Adblock Plus
- flash block
- Chat Zilla
- Super DragAndGo

#### 开发工具

- visual studio 不必多说
- eclipse 出身于Java但他不单单是Java开发工具，目前他已经是一个通用的语言IDE，我一直用eclipse写PHP,Python,Perl还有Docbook XML
- TortoiseSVN 版本控制工具.
- WinMerge 文件差异比较与合并 , Beyond Compare 我用过最好的比较合并工具。







# 第 6 章 project

## 目录

### [1. Project directory](#)

## 1. Project directory

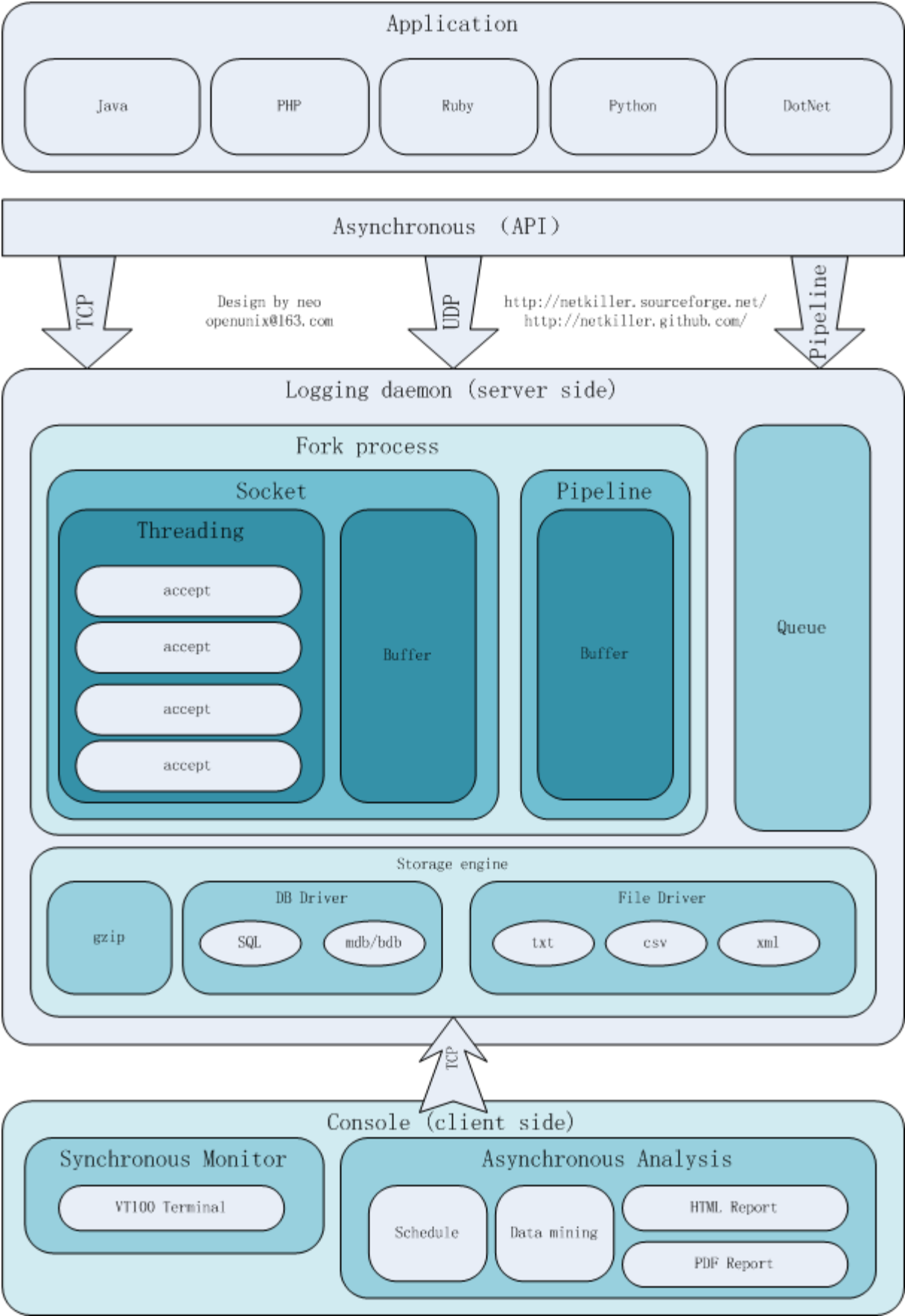
目录规划原则，临时文件分离，日志分离，配置文件分离,这样有利于在负载均衡环境中克隆节点。

- 临时目录 /www/tmp/
- 日志 目录/www/log/
- 配置文件/www/conf

# 第 7 章 Logging

## 目录

### [1. Remote Syslog](#)



# 1. Remote Syslog

syslog

```
syslogd
-----
main {
    fork(){
        socket()

        threading{
            while buffer() {
                queue.add()
            }

        }

        socket.close()
    }
}

storage_engine
-----
main(){
    queue()
    format()
    save()
}

save(){
    open()
    write()
    close()
}
```



# 第 8 章 RDBMS 数据库设计

## 目录

- [1. 数据字典](#)
- [2. 用户帐号表](#)
- [3. 分类表设计](#)
  - [3.1. 树形分类表](#)
  - [3.2. 多对多分类](#)
  - [3.3. 快速检索子分类设计](#)
- [4. 文章表设计](#)
  - [4.1. 分区表设计](#)
  - [4.2. Title性能优化](#)
- [5. 评论表](#)
- [6. 记录点击率，阅读次数，及评分表](#)
- [7. 产品属性表](#)
  - [7.1. 简单实现](#)
  - [7.2. 实现属性组管理](#)
  - [7.3. 可编辑属表](#)
- [8. 国际化语言表](#)
- [9. Workflow](#)
- [10. 内容版本控制](#)
- [11. Sharding](#)
  - [11.1. horizontal](#)
  - [11.2. vertical](#)

仅供参考

# 1. 数据字典

我比建议使用传统的《数据字典》，我的做法是E-R图加数据库注释

注释伴随表，视图，触发器，过程等等，便于维护



## 2. 用户帐号表

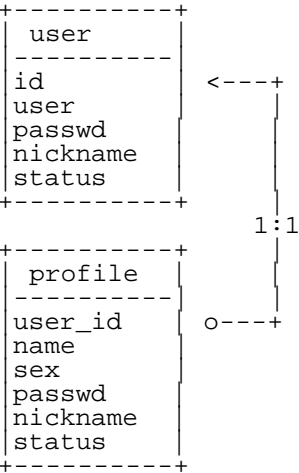
用户帐号或通行证系统设计，下面以我的数库为例讲解。

我一般使用两个表 passport，profile 完成网站会员系统。

首先说说passport表，你也要以使用user或member等等命名，这个表设计尽可能地简单，不要使用过多字段。仅保存登录所必须用到的字段，如user,password,nickname,email... 登录帐号和密码做复合索引。

然后是profile表，这个表与passport是1:1关系，保存用户详细信息

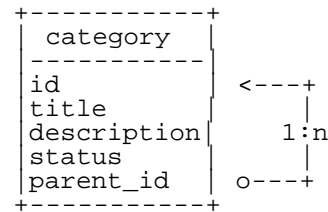
这样设计可以保证海量用户登录时的速度。





### 3. 分类表设计

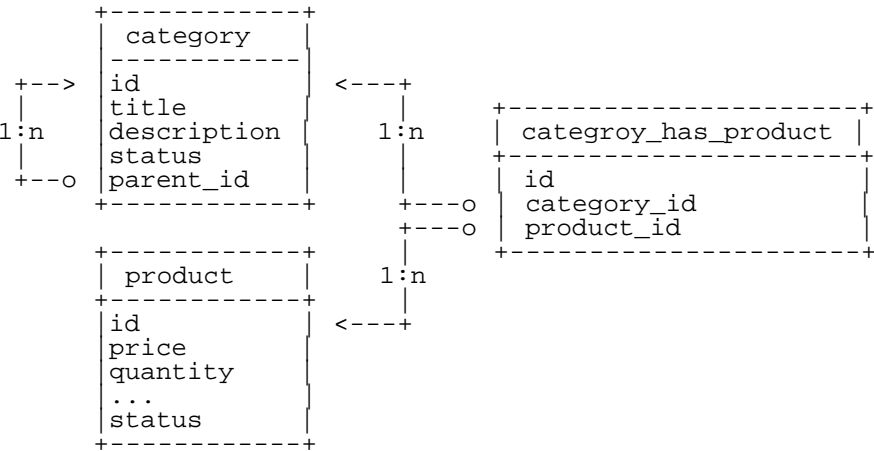
#### 3.1. 树形分类表



```
CREATE TABLE `category` (
  `id` SMALLINT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(10) NOT NULL,
  `description` VARCHAR(255) NULL,
  `status` ENUM('enable','desable') NOT NULL DEFAULT 'enable',
  `parent_id` SMALLINT(10) UNSIGNED NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`),
  CONSTRAINT `FK1` FOREIGN KEY (`parent_id`) REFERENCES `category` (`id`)
)
COMMENT='goods category'
ENGINE=InnoDB
ROW_FORMAT=DEFAULT
```

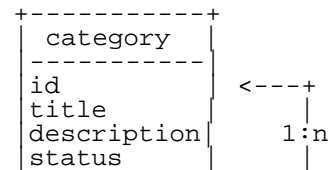
#### 3.2. 多对多分类

多对多分类,主要用于满足，一个产品/文章属于多个分类的需求。



#### 3.3. 快速检索子分类设计

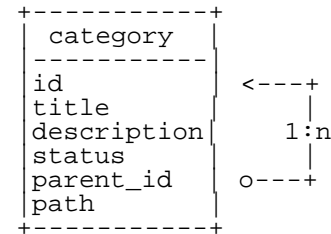
上面我刚刚讲过怎样实现“不限子树的分类树”，我们可以实现不限层次的无线分类表。



```
|parent_id | o---+
+-----+
```

问题出来了，当我需要读取一个分类（任意分类）下的所有子分类，怎样实现，很多人会说用“递归”。当然“递归”可是现实我们的需求，在几百个分类的项目中，使用递归也不是不可以的，但是当数量非常庞大时怎么办？

当然有更好的解决方案，请看下面



category						
id	name	description		status	parent_id	path
1	中国	中华人民共和国				Y NULL 1/
4	广东省	广东省		Y		1 1/
5	深圳市	NULL		Y	4	1/4/5
6	宝安区	NULL		Y	5	1/4/5/6
7	龙华镇	NULL		Y	6	1/4/5/6/7

```
CREATE TABLE `category` (
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '分类ID',
  `name` VARCHAR(50) NOT NULL COMMENT '分类名称',
  `description` VARCHAR(200) NULL DEFAULT NULL COMMENT '分类描述',
  `status` ENUM('Y','N') NOT NULL DEFAULT 'Y' COMMENT '分类状态有继承性',
  `parent_id` INT(10) NULL DEFAULT '1' COMMENT '分类父ID',
  `path` VARCHAR(255) NOT NULL COMMENT '分类递归路径索引',
  INDEX `PK` (`id`),
  INDEX `relation` (`id`, `parent_id`),
  INDEX `FK_category_category` (`parent_id`),
  INDEX `path` (`path`)
)
COMMENT='分类表'
ENGINE=InnoDB
ROW_FORMAT=DEFAULT
AUTO_INCREMENT=0

insert into category(`name`,`description`,`status`,`parent_id`,`path`) values('中国','中华人民共和国','Y',null,'1/')
```

```
ALTER TABLE `category`
ADD CONSTRAINT `FK_category_category` FOREIGN KEY (`parent_id`) REFERENCES `category`
(`id`)
```

抽取广东子树

```
select * from category where path like '1/4%';
```

```
mysql> select * from category where path like '1/4%';
+-----+
| id | name      | description | status | parent_id | path |
+-----+
| 4  | 广东省   | 广东省     | Y      | 1         | 1/4  |
| 5  | 深圳市   | NULL       | Y      | 4         | 1/4/5 |
| 6  | 宝安区   | NULL       | Y      | 5         | 1/4/5/6 |
| 7  | 龙华镇   | NULL       | Y      | 6         | 1/4/5/6/7 |
+-----+
4 rows in set (0.00 sec)
```

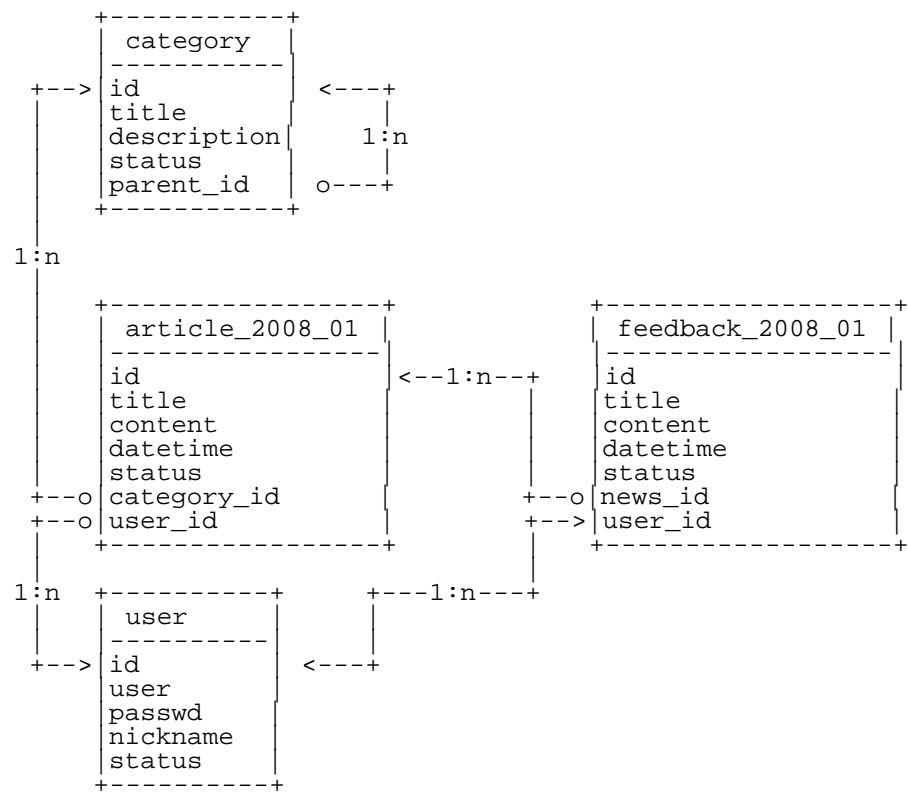






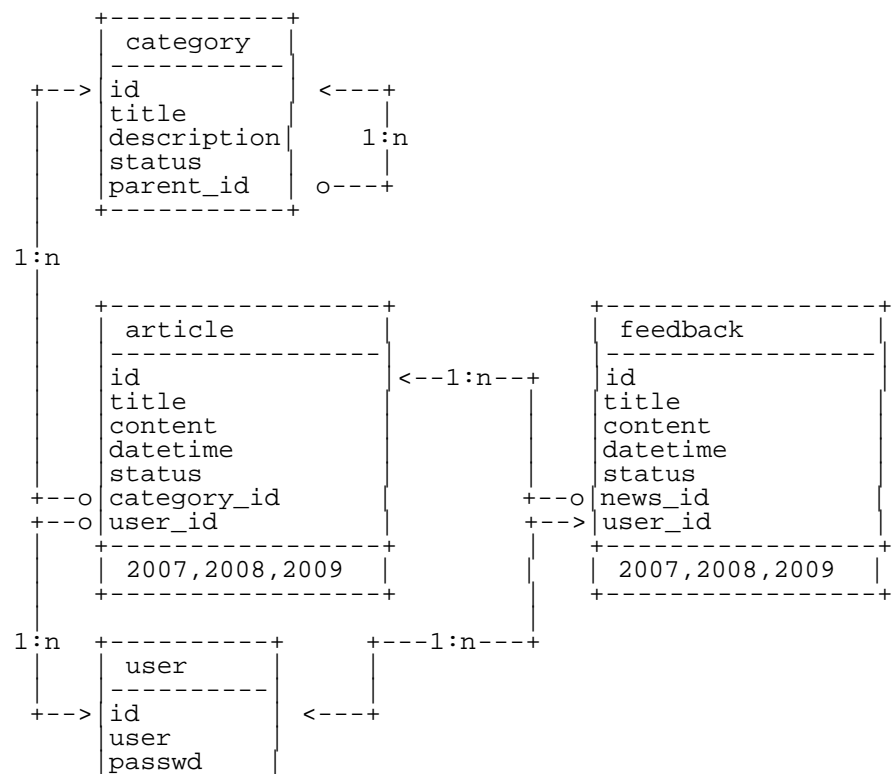
## 4. 文章表设计

看具体情况，拆分表，可按“日”，“月”，“年”等等



### 4.1. 分区表设计

分区表可以通过表空间，等等技术实现，优点是解决了Union查询问题，保证了数据的一致性。



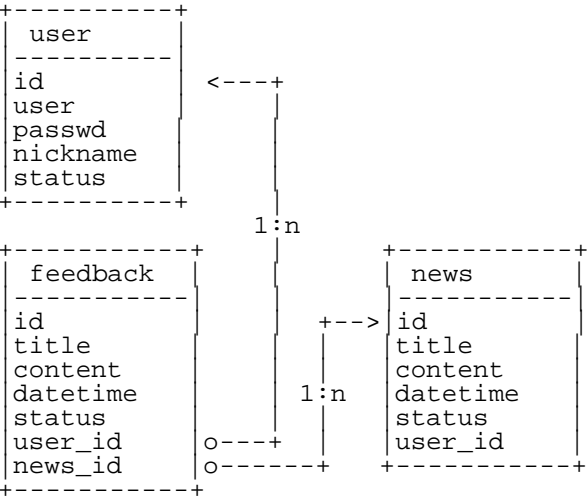
nickname
status

#### 4.2. Title性能优化

显示title前20个汉字并在后尾添加省略号。

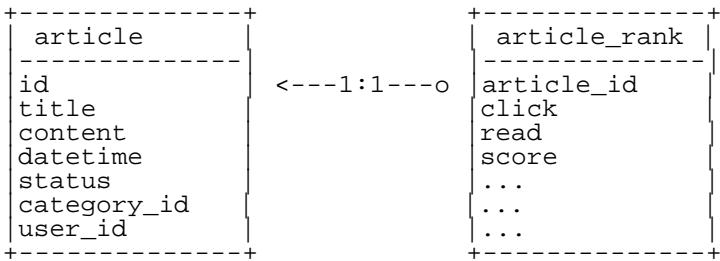


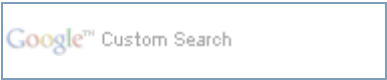
5. 评论表





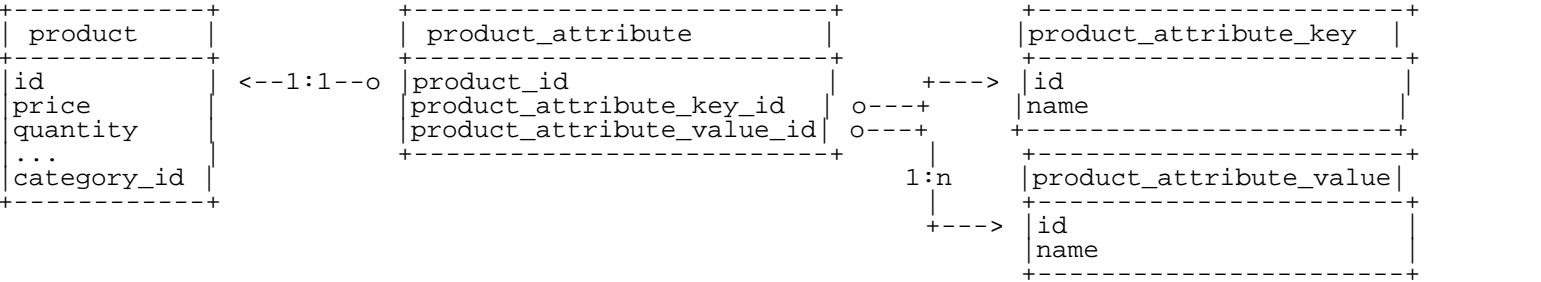
6. 记录点击率，阅读次数，及评分表





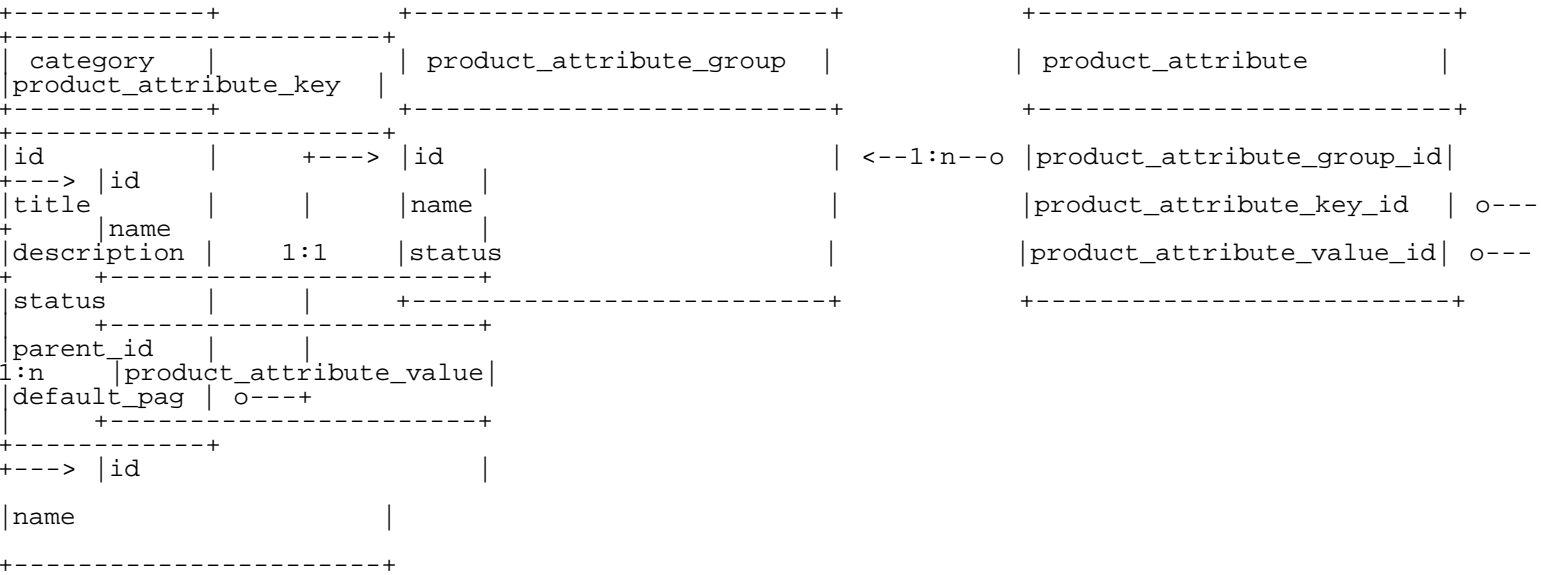
7. 产品属性表

7.1. 简单实现



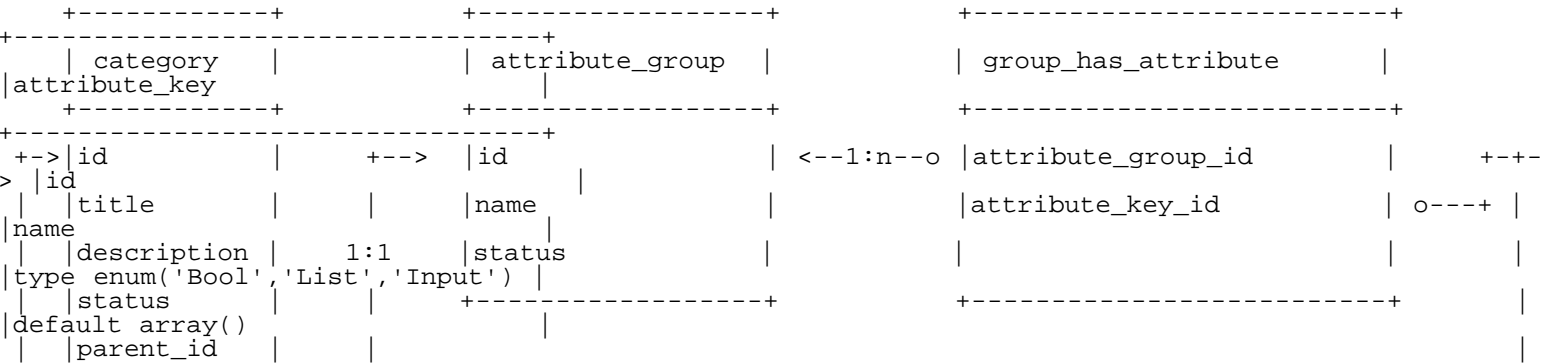
7.2. 实现属性组管理

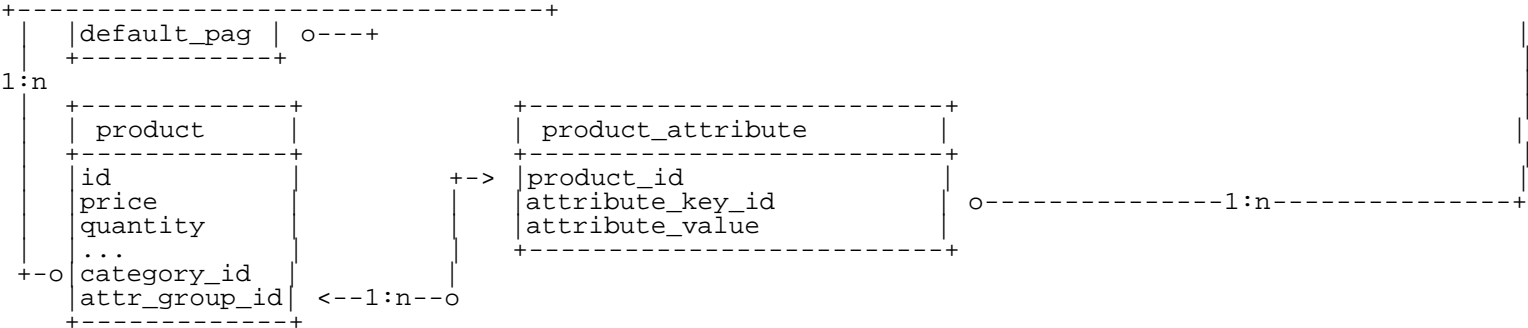
product attribute group



7.3. 可编辑属表

product attribute group

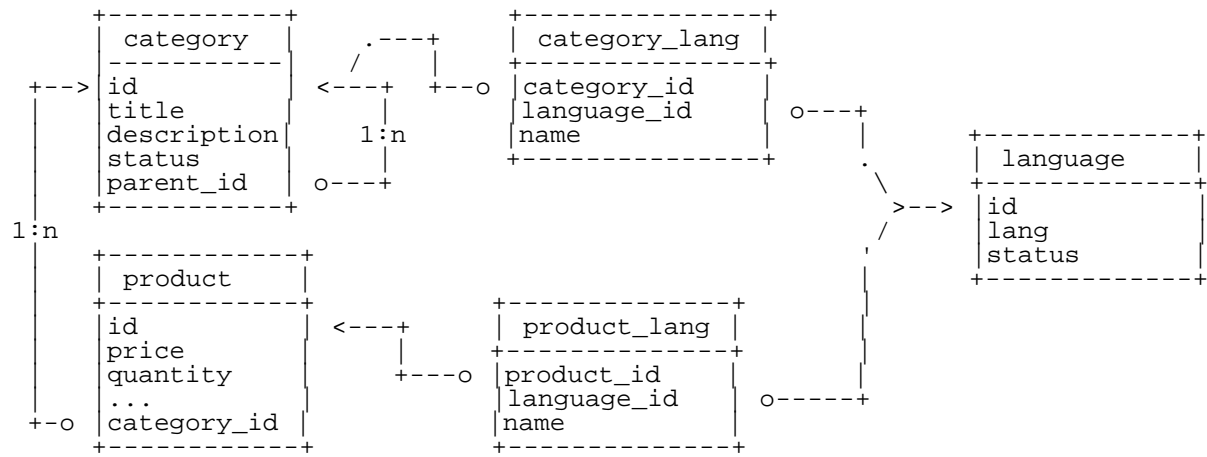




product_attribute_key			
1	color	list	red,green,blue
2	sex	bool	Female,Male
3	qty	input	



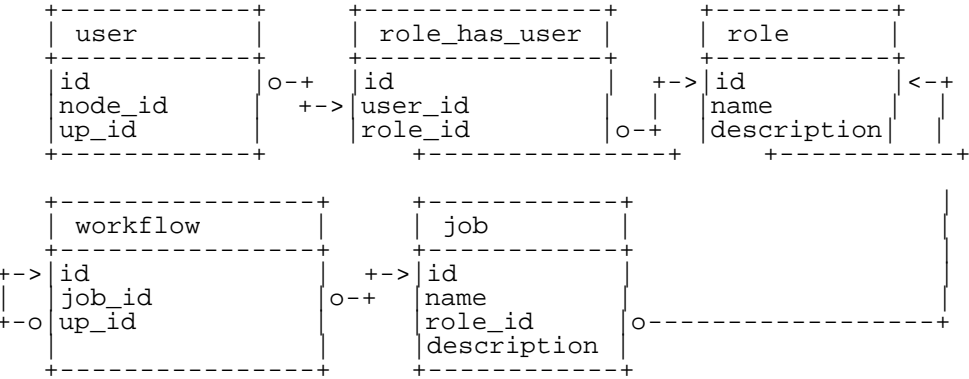
8. 国际化语言表







9. Workflow





## 10. 内容版本控制

主表

```
CREATE TABLE `article` (  
  `article_id` MEDIUMINT(8) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `cat_id` SMALLINT(5) NOT NULL DEFAULT '0',  
  `title` VARCHAR(150) NOT NULL DEFAULT '',  
  `content` LONGTEXT NOT NULL,  
  `author` VARCHAR(30) NOT NULL DEFAULT '',  
  `keywords` VARCHAR(255) NOT NULL DEFAULT '',  
  PRIMARY KEY (`article_id`),  
  INDEX `cat_id` (`cat_id`)  
)  
ENGINE=MyISAM  
ROW_FORMAT=DEFAULT  
AUTO_INCREMENT=1
```

本版控制表，用于记录每次变动

```
CREATE TABLE `article_history` (  
  `id` MEDIUMINT(8) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `article_id` MEDIUMINT(8) UNSIGNED NOT NULL,  
  `cat_id` SMALLINT(5) NOT NULL DEFAULT '0',  
  `title` VARCHAR(150) NOT NULL DEFAULT '',  
  `content` LONGTEXT NOT NULL,  
  `author` VARCHAR(30) NOT NULL DEFAULT '',  
  `keywords` VARCHAR(255) NOT NULL DEFAULT '',  
  PRIMARY KEY (`id`),  
  INDEX `article_id` (`article_id`)  
)  
ENGINE=MyISAM  
ROW_FORMAT=DEFAULT  
AUTO_INCREMENT=1
```

版本控制触发器

```
DROP TRIGGER article_history;  
  
DELIMITER //  
CREATE TRIGGER article_history BEFORE update ON article FOR EACH ROW  
BEGIN  
  INSERT INTO article_history SELECT * FROM article WHERE article_id = OLD.article_id;  
END; //  
DELIMITER;
```



## 11. Sharding

Sharding是近几年提出的概念，可以做分表，分库切割，通过hash值定位。但都存在一个问题，数据连续性，索引无法跨表。

Oracle 在8.x中就支持分区功能，MySQL在5.1.x中也是闲类似功能，PostgreSQL 因存储结构设计的较好，基本不需要做分区。

### 11.1. horizontal

```
ALTER TABLE `goods` DROP INDEX `goods_sn_2`;
ALTER TABLE goods PARTITION BY RANGE (goods_id) (
    PARTITION p0 VALUES LESS THAN (10000),
    PARTITION p1 VALUES LESS THAN (20000),
    PARTITION p2 VALUES LESS THAN (30000),
    PARTITION p3 VALUES LESS THAN (40000),
    PARTITION p4 VALUES LESS THAN MAXVALUE
);

ALTER TABLE goods PARTITION BY HASH(goods_id) PARTITIONS 10;

ALTER TABLE goods PARTITION BY KEY (is_on_sale) PARTITIONS 2;

ALTER TABLE goods PARTITION BY HASH(YEAR(FROM_UNIXTIME(add_time))) PARTITIONS 4;
```

### 11.2. vertical

[Home](#) | [Mirror](#) | [Search](#)



# 第9章 页面设计 Design

## 目录

[1. XHTML+DIV+CSS](#)

[2. 页面结构设计](#)

[2.1. Home page \(首页\)](#)

[2.2. 导航栏](#)

[2.3. Left Bar](#)

[2.4. 区块设计 Block](#)

[3. 表格](#)

[4. 图片优化](#)

[4.1. onMouseOver/onMouseOut](#)

[4.2. 使用一幅图片处理BLOCK四角](#)

[4.3. 图片用背景图代替img 标记](#)

[5. HTML嵌入图片](#)

[6. 页面内容安全](#)

[6.1. 禁止鼠标右键](#)

[6.2. 禁止复制剪切 及粘贴](#)

[7. html.css 有效性检查 Validation](#)

最近几年，随着业界越来越关注XHTML+CSS的标准化设计，一个新兴职业已经诞生，这就是“网站重构师”，这个新兴职业人才紧缺，他们主要的职责是将HTML+Table+Javascript的架构向XHTML+CSS+Ajax迁移。

## 1. XHTML+DIV+CSS

为何使用表格排版是不明智的选择？为什么要选择DIV+CSS？

首选我来说说表格排版，表格排版也是有好处的，一是排版速度快，二是兼容性比CSS好。做为一般的小网站还是比较适合的，如果在大型网站使用表格就不太合适。表格必须定义很多属性如width="100%" border="0" cellpadding="0" cellspacing="0"，并且有时候tr标签显得多余。

### 例 9.1. 表格排版范例

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Table Example</title>
</head>

<body>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td>Logo</td>
    <td>Banner</td>
  </tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td>Home</td>
    <td>News</td>
    <td>Contact</td>
    <td>>About</td>
  </tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td><table width="100%" border="0" cellspacing="0" cellpadding="0">
      <tr>
        <td>Top 10 </td>
      </tr>
      <tr>
        <td>xxxxxxxxx</td>
      </tr>
      <tr>
        <td>xxxxxxxxx</td>
      </tr>
      <tr>
        <td>xxxxxxxxx</td>
      </tr>
      <tr>
        <td>xxxxxxxxx</td>
      </tr>
      <tr>
        <td>xxxxxxxxx</td>
      </tr>
    </table>
    <table width="100%" border="0" cellspacing="0" cellpadding="0">
      <tr>
        <td>Link </td>
      </tr>
      <tr>
        <td>xxxxxxxxx</td>
      </tr>
      <tr>
        <td>xxxxxxxxx</td>
      </tr>
      <tr>
        <td>xxxxxxxxx</td>
      </tr>
      <tr>
        <td>xxxxxxxxx</td>
      </tr>
      <tr>
        <td>xxxxxxxxx</td>
      </tr>
    </table></td>
    <td><table width="100%" border="0" cellspacing="0" cellpadding="0">
      <tr>
        <td align="center">Article Title </td>
      </tr>
      <tr>
        <td><p>Contect</p>
          <p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
          <p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
          <p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
          <p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p></td>
        </tr>
        <tr>
          <td>Feedback</td>
        </tr>
      </table></td>
    </tr>
  </table>
  <table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
      <td align="center">Copyright XXXX </td>
    </tr>
  </table>
  <p>&nbsp;</p>
  <p>&nbsp;</p>
  <p>&nbsp;</p>
</body>
</html>
```

你可以对比上面看看div+css是如何规划版面，并且css很多定义是可以重用的。

例 9.2. XHTML+DIV+CSS排版范例

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>hello world</title>
<style type="text/css">
<!--
body{
    width: 795px;
}
div1{
    border-color: #119EBA;
    border-width: 1px;
    border-style: solid;
    margin: 5px;
}
#header{
}
#logo, #banner{
    float:left;
    height: 75px;
}
#nav{
    clear:both;
}
#nav ul {
    list-style-type:none;
    margin: 0px;
    padding:0px;
}
#nav ul li{
    float:left;
    width: 100px;
}
#main{ clear:both;}
#main #left {
    float:left;
    width: 30%;
}
#main #right {
    float:right;
    width: 70%;
}
.box{}
.box h2 {
    margin: 0;
    padding: 0px;
}
.box a { display:block;}
#footer{ clear:both}
#footer #copyright {
    text-align:center;
}

.article {
    border-color: black;
    border-width: 1px;
    border-style: solid;
    margin: 5px;
    padding: 10px;
}
.article .article_title{
    font-size: 24px;
    font-weight:bold;
    text-align:center;
}
.article .article_content{ font-size:10px;}
-->
</style>
</head>

<body>

<div id="header">
    <div id="logo"> Logo </div>
    <div id="banner"> Banner </div>
</div>

<div id="nav">
    <ul>
        <li><a href="#"> Home </a></li>
        <li><a href="#"> News </a></li>
        <li><a href="#"> Person </a></li>
        <li><a href="#"> Group </a></li>
        <li><a href="#"> Network </a></li>
    </ul>
</div>

<div id="main">
    <div id="left">
        <div class="box">
            <h2>title</h2>
            <a href="#"> link </a>
            <a href="#"> link </a>
            <a href="#"> link </a>
        </div>
    </div>
    <div id="right">
        <div class="box">
            <h2>title</h2>
            <a href="#"> link </a>
            <a href="#"> link </a>
            <a href="#"> link </a>
        </div>
    </div>
</div>

<div id="footer">
    <div id="copyright">
        Copyright © 2006-2010
    </div>
</div>
</body>
</html>
```

```

        <div class="box">
            <h2>title</h2>
            <a href="#"> link </a>
            <a href="#"> link </a>
            <a href="#"> link </a>
        </div>
    </div>
    <div id="right">
        <div class="article">
            <div class="article_title">
                Article Title
            </div>
            <div class="article_content">
                <p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
                <p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
                <p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
                <p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
            </div>
        </div>
    </div>
</div>

<div id="footer">
    <div id="copyright"> Copyright Neo Chan</div>
</div>

</body>
</html>
```

上面例子我们可以看到div与table相比所使用的标签更少，无形中给网站减了肥。

CSS 的class,id 名称定义规范：

- 1. 一定要简单，可读例如header， footer
- 2. 对于在页面中不重复，自始至终只出现一次可定义为id， 例如id="header", id="footer"
- 3. 对于在页面中经常重复出现的，可定义为class， 例如id="article\_block", id="news\_block"

### 注意

不要使用HTML属性，尽量使用css。 herf,src,class,id等属性除外。

下面是一个例子

```
<font color="red" size="12" face="Arial, Helvetica, sans-serif" >Hello workd</font>
```

你应该使用CSS实现,如果能使用CSS实现尽量不要多用一条HTML和属性。

```
<style type="text/css">
.hello{
    color:red;
    font-size:12px;
    font-family:Arial, Helvetica, sans-serif;
}
</style>
<div id="hello">Hello workd</font>
```



## 2. 页面结构设计

页面结构从上到下依次是

- header 主要包括导航，登录，Logo, Banner
- body 网站主要内容，并且还可以分为左右两栏，左中右三栏。
- footer 导航，版权

header,footer将显示在所有页面，一般很少改动。

### 2.1. Home page (首页)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Title</title>
</head>

<body>

<div id="header">
    <div id="logo"></div>
    <div id="banner"></div>
    <div id="nav"></div>
</div>

<div id="page1">
    <div class="left_nav"></div>
    <div class="right"></div>
</div>

<div id="page2">

</div>

<div id="page3">

</div>

<div id="footer">
    <div id="footer_nav"></div>
    <div id="copyright"></div>
</div>

</body>
</html>
```

page1 打开首页看到的第一屏页面，page2，page3需要按翻页键

不要将 page1，page2，page3 放到一个DIV中

### 2.2. 导航栏

```
<table>
  <tr>
    <td>Home</td>
```



```

        <td>News</td>
        <td>About</td>
        <td>Contact</td>
    </tr>
</table>

<div id="nav">
    <ul>
        <li><a href="#"> Home </a></li>
        <li><a href="#"> News </a></li>
        <li><a href="#"> Person </a></li>
        <li><a href="#"> Group </a></li>
        <li><a href="#"> Network </a></li>
    </ul>
</div>
```

2.3. Left Bar

```

<div id="left">
    <div class="box">
        <h2>title</h2>
        <a href="#"> link </a>
        <a href="#"> link </a>
        <a href="#"> link </a>
    </div>

    <div class="box">
        <h2>title</h2>
        <a href="#"> link </a>
        <a href="#"> link </a>
        <a href="#"> link </a>
    </div>
</div>
```

2.4. 区块设计 Block

网站经常用一些方块规划版面。

- 一种是矩形方框
- 另一种是有标题，标题下方是矩形方框
- 现在流行的是标题栏有多个选项卡，标题下方是矩形方框，当选择不同标题时，矩形方框中的内容随之改变。

传统方法如下：

例 9.3. 例子

table block example

```

<table>
    <tr>
        <td>
            内容
        </td>
    </tr>
</table>
```

div+css block example

```

<div class="simple_box">
    内容
</table>
```

例 9.4. 例子

table title block example

```
<table>
  <tr>
    <td>Top 10</td>
  </tr>
  <tr>
    <td>
      <table>
        <tr>
          <td>No.1</td>
        </tr>
        <tr>
          <td>No.2</td>
        </tr>
        <tr>
          <td>No.3</td>
        </tr>
        <tr>
          <td>No.n</td>
        </tr>
      </table>
    </td>
  </tr>
</table>
```

div+css title block example

```
<div class="title_block">
  <h2>
    Title
  <h2>
    <div>
      Content
    </div>
</div>
```

使用dl标记实现

```
<dl class="title_block">
  <dt>Title<dt>
  <dd>
    Content
  </dd>
</dl>
```



## 3. 表格

这里的表格不是指用于排版，而是表格数据。

```
<style type="text/css">
.hello{
    width: 100%;
}
.hello tr{
}
.hello td{
}
</style>
<table class="mytable">
    <tr>
        <td></td>
    </tr>
</table>
```



## 4. 图片优化

### 4.1. onMouseOver/onMouseOut

我们在网站冲浪常常看会看到很多图片按钮，当鼠标入上去或鼠标移开图片会随之改变，这个的按钮至少需要三张小图片才能实现这样的功能。

我先说说这样做的缺点

- 三张图片，你的浏览器会启动三个线程链接你的图片服务器，不划算。
- 一旦其中一幅图片下载过程中中断，用户当把鼠标放到按钮上时，可能会出现一个红叉叉。
- 图片太多不好维护，易产生垃圾，占用磁盘空间，linux ext2 一个空文件占用1024K

最优方法是使用一张图片，将三幅图片平行或垂直排开，放到一幅图片中，然后使用CSS控制显示你需要的部分。

### 4.2. 使用一幅图片处理BLOCK四角

[corner.gif](#)

stylesheet

```
<style type="text/css">
<!--

.clear { clear: both; height: 0; font-size: 0; line-height: 0; zoom: 1 }

.containerPlain {
    background-color: #fff;
    border-right: 1px solid #cacaca;
    border-left: 1px solid #cacaca;
    padding: 0 3px;
}

.left_top_corner {
    display: block;
    width: 4px;
    height: 4px;
    background-position: top left;
    float: left;
    font-size: 0;
}

.right_top_corner {
    display: block;
    width: 4px;
    height: 4px;
    background-position: top right;
    float: right;
    font-size: 0;
}

.left_bottom_corner {
    display: block;
    width: 4px;
    height: 4px;
    background-position: bottom left;
    float: left;
    font-size: 0;
}
```

```
        background-position: bottom left;
        float: left;
        font-size: 0;
    }

    .right_bottom_corner {
        display: block;
        width: 4px;
        height: 4px;
        background-position: bottom right;
        float: right;
        font-size: 0;
    }
    .left_bottom_corner, .right_bottom_corner ,
    .left_top_corner, .right_top_corner{
        background-image: url(corners/corner.gif);
    }

    .middle_top_line {
        display: block;
        float: left;
        height: 3px;
        line-height: 0;
        border-top: 1px solid #cacaca;
    }

    .middle_bottom_line {
        display: block;
        float: left;
        height: 3px;
        border-bottom: 1px solid #cacaca;
        font-size: 0;
    }

    .middle_top_line, .middle_bottom_line {
        width: 167px;
    }

-->
</style>
```

HTML

```
<div style="width:175px;">
    <span class="left_top_corner"></span> <span class="middle_top_line"></span> <span
class="right_top_corner"></span>
    <div class="containerPlain">
        You Content
    </div>
    <span class="left_bottom_corner"></span> <span class="middle_bottom_line"></span> <span
class="right_bottom_corner"></span>
</div>
```

下面是一个更复杂的例子

- [corner.gif](#)
- [block\\_title\\_left.gif](#)
- [block\\_title\\_right.gif](#)

stylesheet

```
<style type="text/css">
<!--

.clear { clear: both; height: 0; font-size: 0; line-height: 0; zoom: 1 }

.containerPlain {
    background-color: #fff;
    border-right: 1px solid #cacaca;
    border-left: 1px solid #cacaca;
    padding: 0 3px;
    clear: both;
}

.left_top_corner {
    display: block;
    width: 4px;
    height: 4px;
    background-position: top left;
    float: left;
    font-size: 0;
}

.right_top_corner {
    display: block;
```

```
width: 4px;
height: 4px;
background-position: top right;
float: right;
font-size: 0;
}

.left_bottom_corner {
display: block;
width: 4px;
height: 4px;
background-position: bottom left;
float: left;
font-size: 0;
}

.right_bottom_corner {
display: block;
width: 4px;
height: 4px;
background-position: bottom right;
float: right;
font-size: 0;
}

.left_bottom_corner, .right_bottom_corner ,
.left_top_corner, .right_top_corner{
background-image: url(corners/corner.gif);
}

.middle_top_line {
display: block;
float: left;
height: 3px;
line-height: 0;
border-top: 1px solid #cacaca;
}

.middle_bottom_line {
display: block;
float: left;
height: 3px;
border-bottom: 1px solid #cacaca;
font-size: 0;
}

.middle_top_line, .middle_bottom_line {
width: 167px;
}

.block_title {
line-height: 26px;
height: 26px;
background-image: url(corners/block_title_left.png);
background-repeat: no-repeat;
padding-left: 10px;
font-size: 13px;
font-weight: bold;
background-color: #dddbdc;
}

.block_title_right {
display: block;
background-image: url(corners/block_title_right.png);
background-repeat: no-repeat;
background-postition: right;
float: right;
width: 4px;
height: 26px;
}
-->
</style>
```

HTML

```
<div style="width:175px;">
  <span class="left_top_corner"></span> <span class="middle_top_line"></span> <span
class="right_top_corner"></span>
  <div class="containerPlain">
    <div class="block_title">
      <span class="block_title_right"></span> Title
    </div>
    <div style="padding: 10px 7px 7px 7px">
      Content
    </div>
  </div>
  <span class="left_bottom_corner"></span> <span class="middle_bottom_line"></span> <span
class="right_bottom_corner"></span>
</div>
```

4.3. 图片用背景图代替img 标记

图片用背景图代替<img src="">

---

<a href="#">上一页</a>	<a href="#">上一级</a>	<a href="#">下一页</a>
3. 表格	<a href="#">起始页</a>	5. HTML嵌入图片



## 5. HTML嵌入图片

```

```





## 6. 页面内容安全

### 6.1. 禁止鼠标右键

修改body标签，加入onContextMenu="return false" onSelectStart="return false"。

```
<body bgColor="#FFFFFF" onContextMenu="return false" onSelectStart="return false">
```

### 6.2. 禁止复制剪切 及粘贴

禁止拷贝文字：

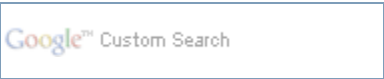
```
<input name="textfield" type="text" value="不能复制里面的字" oncopy="return false;" oncut="return false;" onpaste="return false">
```



## 7. html,css 有效性检查 Validation

有效性检查包括

- 1. Markup Validation
- 2. CSS Validation.
- 3. URL Validation.



# 第 10 章 Framework Design

目录

[1. MVC Framework](#)

[2. HMVC Framework](#)

[3. REST](#)

[3.1. RESTful JSON API](#)

[4. action 分发器](#)

[5. Plugin & Hook 设计与实现](#)

[5.1. 插件管理平台](#)

[5.2. 接口定义](#)

[5.3. 插件](#)

[5.4. 测试](#)

[6. Cache](#)

[6.1. CDN \(Content Delivery Network\) 与反向代理](#)

[6.1.1. CDN接口API](#)

[6.1.2. 方向代理页面过期处理](#)

[6.1.3. 内容版本化](#)

[6.2. HTTP Cache](#)

[6.2.1. 在程序中灵活操作 Cache-Control](#)

[6.2.2. 非程序文件缓存处理](#)

[6.3. Cache 生存时间](#)

[7. Session/Cookie 安全](#)

[7.1. cookie-free domains](#)

[8. i18n 国际化](#)

[8.1. 数组方式](#)

[8.2. 数据库方式](#)

[8.3. 文件文件](#)

[8.4. Gettext](#)

[8.5. 数据结构](#)

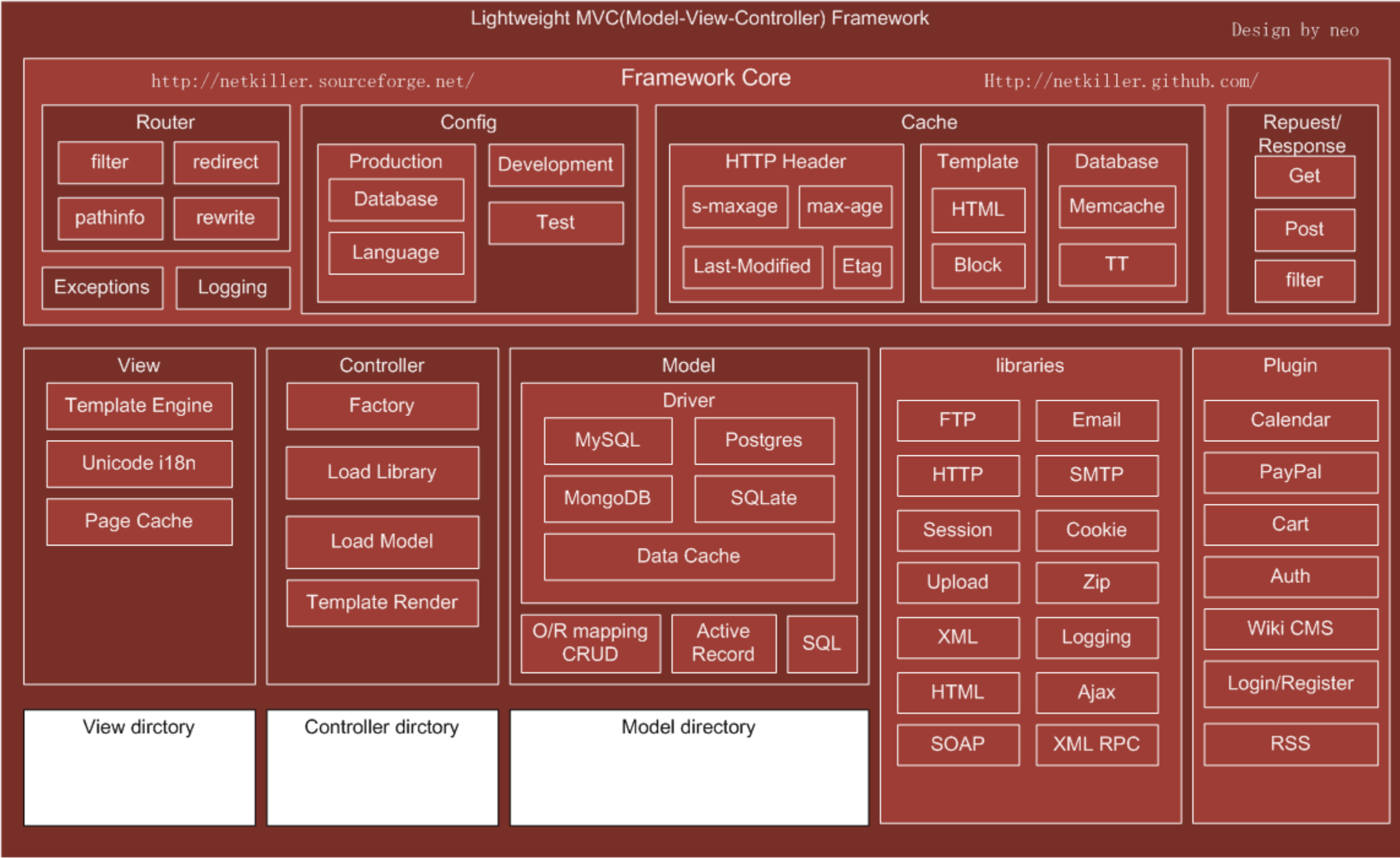
[9. Logging](#)

[10. SEO Module](#)

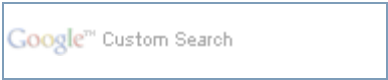
[10.1. robots.txt](#)

[10.1.1. Sitemap in robots.txt](#)

## 1. MVC Framework



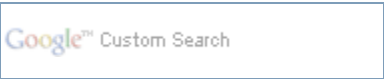




## 2. HMVC Framework

等我有时间在补充

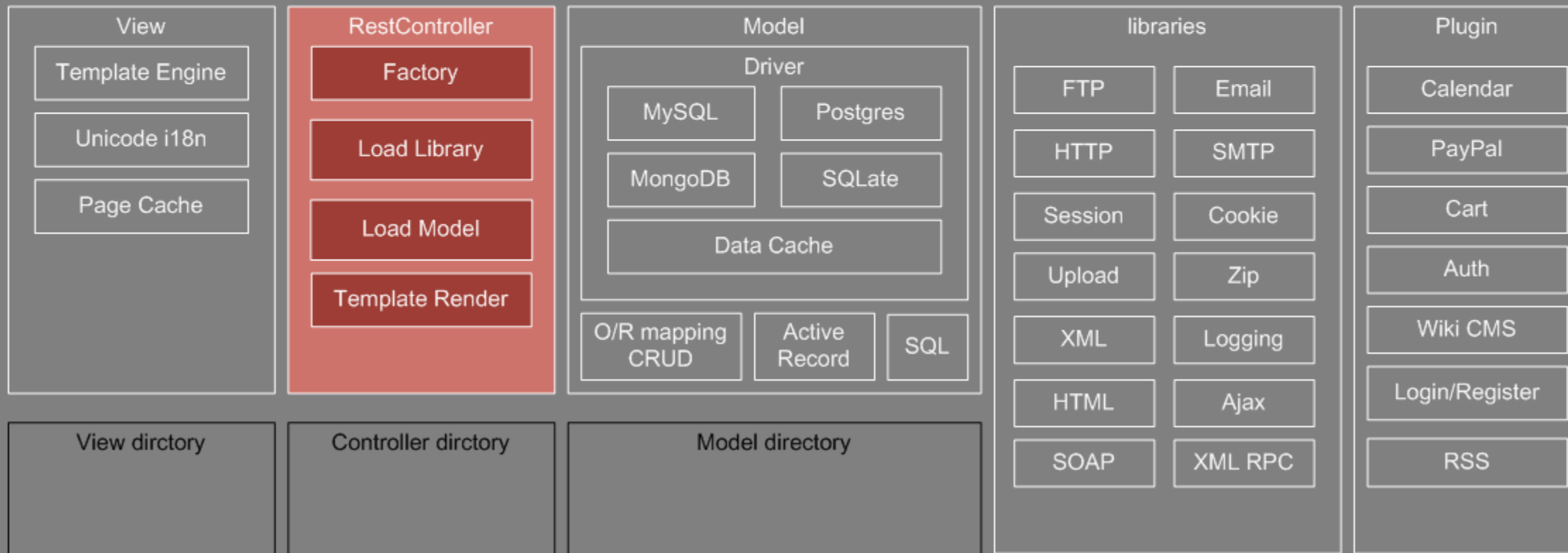
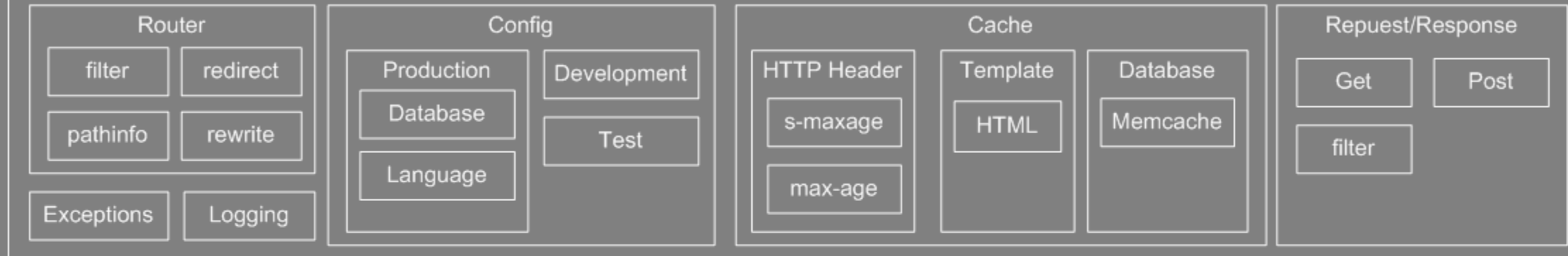




### 3. REST

## MVC(Model-View-Controller) Framework

### Framework Core



### 3.1. RESTful JSON API

[上一页](#)

[上一级](#)

[下一页](#)

2. HMVC Framework

[起始页](#)

4. action 分发器



## 4. action 分发器

```
$action  = $_REQUEST['action'];
$libname = $_REQUEST['lib'];
$special = new Advertize ($libname,$action);
if(method_exists($special, $action)) {
    $special->$action();
}else{
    $special->index();
}
```



## 5. Plugin & Hook 设计与实现

插件系统分为：

插件管理平台

插件探测

插件注册

插件调用

插件注销

### 5.1. 插件管理平台

### 5.1. 插件管理平台

```
<?php
final class Plugin{
    private $plugins          = null;
    private $directory        = 'plugins';
    private $path              = null;
    public function __construct(){
        $this->path = $this->directory.'/';
    }
    public function autoload(){
        $interfaces = scandir($this->directory);
        unset($interfaces[0]);
        unset($interfaces[1]);
        foreach($interfaces as $interface)
        {
            //load all of the plugins
            $file = $this->path . $interface;
            if (@file_exists($file))
            {
                include_once($file);
                $class = basename($interface, ".php");
                if (class_exists($class))
                {
                    $this->$class = new $class($this);
                    $vars = get_class_vars($class);
                    $entity['name'] = $vars['name'];
                    $entity['description'] = $vars['description'];
                    $entity['author'] = $vars['author'];
                    $entity['class'] = $class;
                    $entity['methods'] =
get_class_methods($class);

                    $this->plugins[$class] = $entity;
                }
            }
        }
    }
    public function load($plugin){
        $file = $this->path . $plugin . '.php';
        if (@file_exists($file))
        {
            include_once($file);
            $class = $plugin;
            if (class_exists($class))
            {
                $this->$class = new $class($this);
                $vars = get_class_vars($class);
                $entity['name'] = $vars['name'];
```

```

        $entity['description'] = $vars['description'];
        $entity['author']     = $vars['author'];
        $entity['class']      = $class;
        $entity['methods']    = get_class_methods($class);

        $this->plugins[$class] = $entity;
    }
}

public function show(){
    print_r($this->plugins);
}
}
```

5.2. 接口定义

```
<?php
interface iPlugin
{
    public function test();
}
```

5.3. 插件

```
<?php
final class demo implements iPlugin{
    public static $author      = 'Neo Chen<openunix@163.com>';
    public static $name = 'Demo';
    public static $description = 'Demo Simple';
    public function __construct(){

    }
    public function test(){
        echo 'Hello world!!!';
    }
}
```

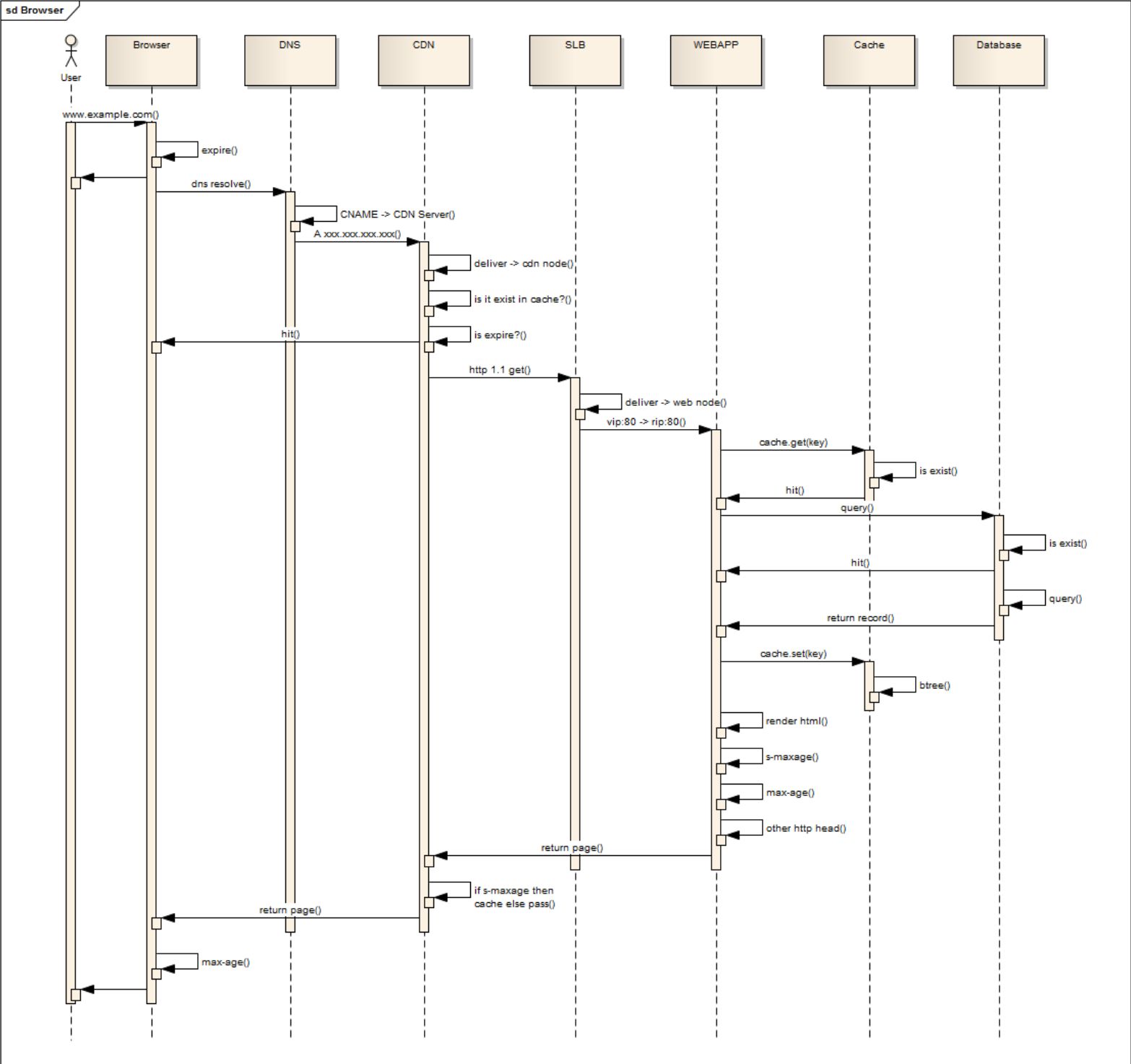
5.4. 测试

```
<?php
function __autoload($class_name) {
    require_once('library/'.$class_name . '.php');
}

//include_once('library/Plugin.php');
$plugin = new Plugin();
echo '=====';
$plugin->load('demo');
$plugin->demo->test();
echo '=====';
$plugin->autoload();
$plugin->show();
```

## 6. Cache

### 6.1. CDN (Content Delivery Network) 与反向代理



与CDN有关的开发工作

#### 6.1.1. CDN接口API

CDN 内容更新，一般厂商会提供一个SOAP接口，你可以通过接口刷新你的内容。但接口有限制，不能随意使用，一般是多少秒可以刷新一次，或者一天可以刷新几次

#### 6.1.2. 方向代理页面过期处理

方向代理一般都支持PURGE协议，Squid,Varnish等等向管理端口发送 PURGE 即可是使用页面刷新

PURGE http://netkiller.github.net/index.html

有些方向代理如：Varnish 可以使用正则表达式

同时这些代理服务器都承受管理命令

squid: squidclient

varnish: varnishadm

6.1.3. 内容版本化

例如这样的URL

http://images.example.com/logo.gif  
http://images.example.com/product.jpg

我们可以通过Rewrite或PATHINFO等技术做为静态化。例如首次版本

http://images.example.com/logo.1.gif => logo.gif  
http://images.example.com/product.1.jpg => product.jpg

原图发生变化后，版本递增

http://images.example.com/logo.2.gif => logo.gif  
http://images.example.com/product.2.jpg => product.jpg

就的URL将丢弃

http://images.example.com/logo.1.gif  
http://images.example.com/product.1.jpg

CDN 就回源去下面的URL，并且取到的是新图

http://images.example.com/logo.2.gif  
http://images.example.com/product.2.jpg

6.2. HTTP Cache

Cache-Control

max-age 格式写为: max-age=n, n是以秒为单位，这个值是告知客户端GMT + N 后页面过期，缓存服务器在s-maxage值为空的时候也会使用这个参数的值。

s-maxage 的格式跟max-age一样，只不过他是给缓存服务器使用的。

must-revalidate 这个参数用来告知客户端和缓存服务器，在GET请求的时候必须与源服务器验证实体是否为最新版本。

Cache-Control:max-age=1200,s-maxage=3600

Last-Modified 这个参数提供了实体最近一次被修改的时间。这个名字起得不错，当实体被修改了之后，这个参数也就会被修改。

ETag

ETag是根据内容生成的一段hash字符串，采用信息摘要算法，保证每一个页面有一个唯一字串。

expires

expires 是HTTP 1.0 中定义的，已经不能满足用户的需要在 HTTP 1.1 加入了max-age，建议使用 max-age替代expires

指令	含义
public	可以在任何地方缓存
private	只能被浏览器缓存
no-cache	不能在任何地方缓存
must-revalidate	缓存必须检查更新版本
proxy-revalidate	代理缓存必须检查更新版本
max-age	内容能够被缓存的时期，以秒表示
s-maxage	覆盖共享缓存的max-age设置

在Squid, Varnish, Apache, Lighttpd, Nginx 中都可是实现HTTP Cache-Control推送，每次修改都需要重新加载，不太灵活。

```
ExpiresActive On
ExpiresByType image/gif "access plus 1 month"
ExpiresByType image/png "access plus 1 month"
ExpiresByType image/jpeg "access plus 1 month"
ExpiresByType text/css "access plus 1 month"
ExpiresByType text/javascript "access plus 1 month"
ExpiresByType application/x-javascript "access plus 1 month"
ExpiresByType application/x-shockwave-flash "access plus 1 month"
```

```
server.modules = (
...
"mod_expire",
...
)

$HTTP["url"] =~ "^/images/" {
expire.url = ( " " => "access 30 days" )
}
```

我喜欢自己控制TTL时间，且每个页面单独设置，可以随时调整设置。

6.2.1. 在程序中灵活操作 Cache-Control

在MVC框架中每个控制器下的方法都可以单独操作Cache



```
Class blog extend Controller{
    blog(){
        header('Cache-Control: max-age=28800');
    }
    list(){
        header('Cache-Control: max-age=3600');
    }
    details(){
        header('Cache-Control: max-age=160');
    }
}
```

你还可以封装到Controller中

```
Class blog extend Controller{
    blog(){
        this->cache('28800');
    }
    list(){
        this->cache('3600');
    }
    details(){
        this->cache('160');
    }
}
```

6.2.2. 非程序文件缓存处理

首先做一个Rewrite让程序接管所有图片请求

```
url.rewrite = ( "^(/.)" => "/index.php/$1" )
```

然后程序通过PATHINFO取出图片URL

```
http://images.example.com/your/dir/test.jpg => http://images.example.com/index.php/your/dir/test.jpg
```

程序取出 /your/dir/test.jpg 设置 Content-type 并输出二进制流

详细参考

```
<?php
// Test image.
$images = '/test/foo.png';

$headers = apache_request_headers();

if (isset($headers['If-Modified-Since']) && (strtotime($headers['If-Modified-Since']) == filetime($images))) {
    header('Last-Modified: '.gmdate('D, d M Y H:i:s', filetime($images)).' GMT', true, 304);
} else {
    header('Content-Type: image/png');
    print file_get_contents($fn);
    if (file_exists($images)) {
        header('Last-Modified: '.gmdate('D, d M Y H:i:s', filetime($images)).' GMT', true, 200);
        header("Cache-Control: max-age=3600, must-revalidate");
        header('Content-Length: '.filesize($images));
        header('Content-type: ' .mime_content_type($images));
        flush();
        readfile($images);
        exit;
    }
}
```

javascript 文件也可以使用类似方法处理

```
private function js($file){
    if (file_exists($file)) {
        header("Cache-Control: max-age=3600, must-revalidate");
        header('Content-type: text/javascript');
        flush();
        readfile($file);
        exit;
    }
}
```

6.3. Cache 生存时间

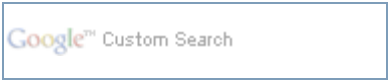
你不必一开始加费劲动机去考虑这个值，当网站运行一段时间后，利用玩站流量数据作为参考，一步一地尝试调整。



# 7. Session/Cookie 安全

## 7.1. cookie-free domains

[Home](#) | [Mirror](#) | [Search](#)



## 8. i18n 国际化

### 8.1. 数组方式

这种方式流行于PHP语言，下面是一个例子

例 10.1. php language package

```
<?php
    $language['hello_world'] = 'hello world !!!'
?>
```

### 8.2. 数据库方式

数据库方式包括

- 1. 其他非关系型收据库 (Berkeley DB)
- 2. 对象/关系型收据库 ORDBMS （mysql）

Berkeley DB 是一个不错的选择，而且相对关系型数据库比较有优势。因为关系型数据库子并发数有限，连接资源很宝贵。

例 10.2. sql table language package

```
select id,key,value from language where country = 'zh-cn' and key = 'hello_world';
```

### 8.3. 文件文件

例如.ini文件

```
news=新闻
top10=前十位
```

### 8.4. Gettext

The gettext functions implement an NLS (Native Language Support) API which can be used to internationalize your PHP applications. Please see the gettext documentation for your system for a thorough explanation of these functions or view the docs at » <http://www.gnu.org/software/gettext/manual/gettext.html>.

## 8.5. 数据结构

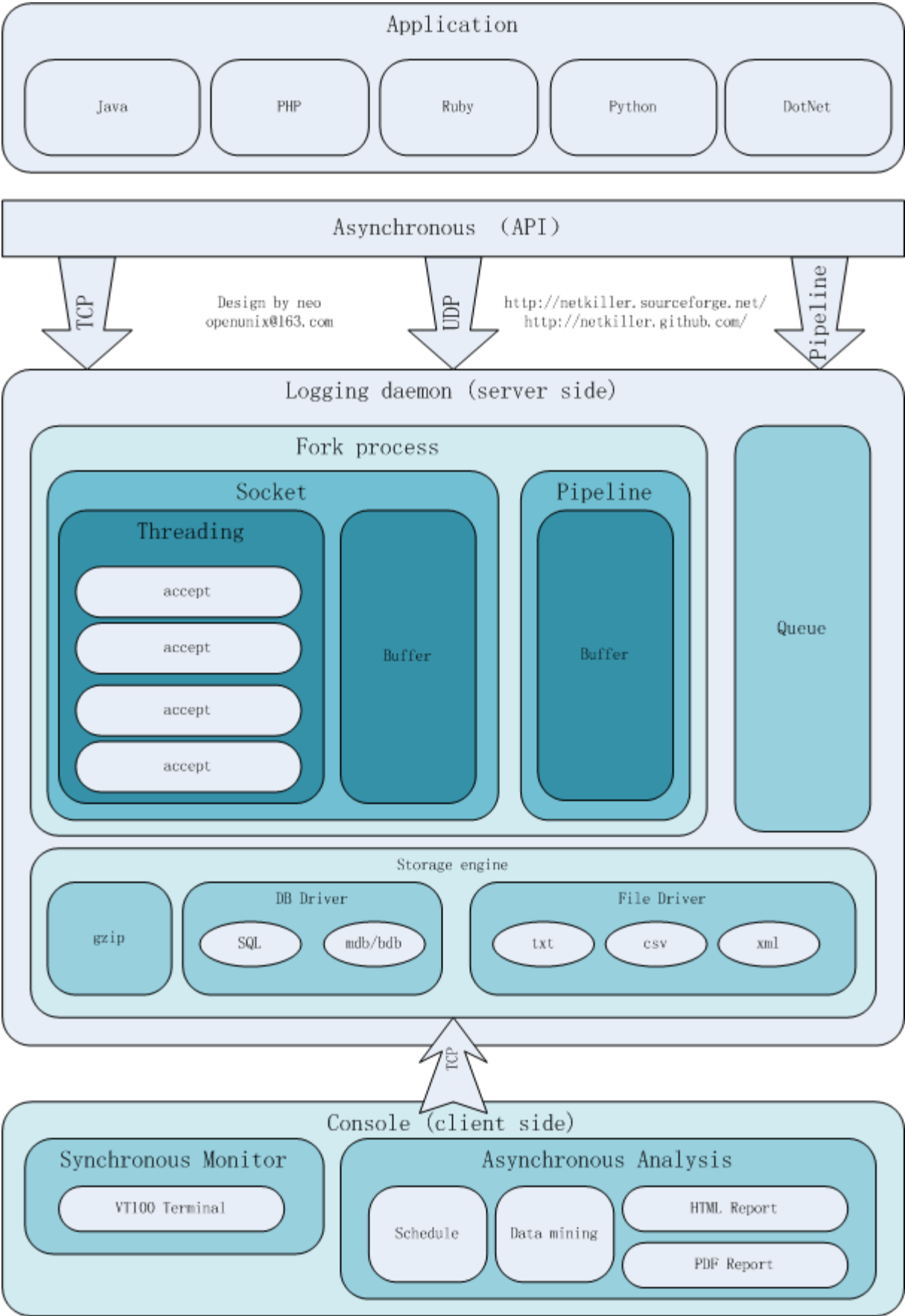
数据结构方式主要包括

- 1. 哈希表 hash table
- 2. 类 class
- 3. 字典 dict
- 4. 图 map

提示

可能会用到序列化

9. Logging





# 10. SEO Module

## 10.1. robots.txt

```
<meta name="robots" content="noarchive">
```

### 例 10.3. example robots.txt

http://www.google.com/robots.txt

```
User-agent: *
Disallow: /search
Disallow: /groups
Disallow: /images
Disallow: /catalogs
Disallow: /catalogues
Disallow: /news
Allow: /news/directory
Disallow: /nwshp
Disallow: /setnewsprefs?
Disallow: /index.html?
Disallow: /?
Disallow: /addurl/image?
Disallow: /pagead/
Disallow: /relpage/
Disallow: /relcontent
Disallow: /imgres
Disallow: /imglanding
Disallow: /keyword/
Disallow: /u/
Disallow: /univ/
Disallow: /cobrand
Disallow: /custom
Disallow: /advanced_group_search
Disallow: /googlesite
Disallow: /preferencessection
Disallow: /setprefs
Disallow: /swr
Disallow: /url
Disallow: /default
Disallow: /m?
Disallow: /m/?
Disallow: /m/blogs?
Disallow: /m/ig
Disallow: /m/images?
Disallow: /m/local?
Disallow: /m/movies?
Disallow: /m/news?
Disallow: /m/news/i?
Disallow: /m/place?
Disallow: /m/setnewsprefs?
Disallow: /m/search?
Disallow: /m/swmloptin?
Disallow: /m/trends
Disallow: /wml?
Disallow: /wml/?
Disallow: /wml/search?
Disallow: /xhtml?
Disallow: /xhtml/?
Disallow: /xhtml/search?
Disallow: /xml?
Disallow: /imode?
Disallow: /imode/?
Disallow: /imode/search?
Disallow: /jsky?
Disallow: /jsky/?
Disallow: /jsky/search?
Disallow: /pda?
Disallow: /pda/?
Disallow: /pda/search?
Disallow: /sprint_xhtml
Disallow: /sprint_wml
Disallow: /pqa
Disallow: /palm
```

Disallow: /gwt/  
Disallow: /purchases  
Disallow: /hws  
Disallow: /bsd?  
Disallow: /linux?  
Disallow: /mac?  
Disallow: /microsoft?  
Disallow: /unclesam?  
Disallow: /answers/search?q=  
Disallow: /local?  
Disallow: /local\_url  
Disallow: /froogle?  
Disallow: /products?  
Disallow: /products/  
Disallow: /froogle\_  
Disallow: /product\_  
Disallow: /products\_  
Disallow: /print  
Disallow: /books  
Disallow: /bkshp?q=  
Allow: /booksrightsholders  
Disallow: /patents?  
Disallow: /patents/  
Allow: /patents/about  
Disallow: /scholar  
Disallow: /complete  
Disallow: /sponsoredlinks  
Disallow: /videosearch?  
Disallow: /videopreview?  
Disallow: /videoprograminfo?  
Disallow: /maps?  
Disallow: /mapstt?  
Disallow: /mapslt?  
Disallow: /maps/stk/  
Disallow: /maps/br?  
Disallow: /mapabcpoi?  
Disallow: /maphp?  
Disallow: /places/  
Disallow: /maps/place  
Disallow: /help/maps/streetview/partners/welcome/  
Disallow: /lochp?  
Disallow: /center  
Disallow: /ie?  
Disallow: /sms/demo?  
Disallow: /katrina?  
Disallow: /blogsearch?  
Disallow: /blogsearch/  
Disallow: /blogsearch\_feeds  
Disallow: /advanced\_blog\_search  
Disallow: /reader/  
Allow: /reader/play  
Disallow: /uds/  
Disallow: /chart?  
Disallow: /transit?  
Disallow: /mbd?  
Disallow: /extern\_js/  
Disallow: /calendar/feeds/  
Disallow: /calendar/ical/  
Disallow: /cl2/feeds/  
Disallow: /cl2/ical/  
Disallow: /coop/directory  
Disallow: /coop/manage  
Disallow: /trends?  
Disallow: /trends/music?  
Disallow: /trends/hottrends?  
Disallow: /trends/viz?  
Disallow: /notebook/search?  
Disallow: /musica  
Disallow: /musicad  
Disallow: /musicas  
Disallow: /musicl  
Disallow: /musics  
Disallow: /musicsearch  
Disallow: /musicsp  
Disallow: /musiclp  
Disallow: /browsersync  
Disallow: /call  
Disallow: /archivesearch?  
Disallow: /archivesearch/url  
Disallow: /archivesearch/advanced\_search  
Disallow: /base/search?  
Disallow: /base/reportbadoffer  
Disallow: /base/s2  
Disallow: /urchin\_test/  
Disallow: /movies?  
Disallow: /codesearch?  
Disallow: /codesearch/feeds/search?  
Disallow: /wapsearch?  
Disallow: /safebrowsing  
Allow: /safebrowsing/diagnostic  
Allow: /safebrowsing/report\_error/  
Allow: /safebrowsing/report\_phish/  
Disallow: /reviews/search?  
Disallow: /orkut/albums  
Disallow: /jsapi  
Disallow: /views?  
Disallow: /c/  
Disallow: /cbk  
Disallow: /recharge/dashboard/car  
Disallow: /recharge/dashboard/static/  
Disallow: /translate\_a/  
Disallow: /translate\_c  
Disallow: /translate\_f  
Disallow: /translate\_static/  
Disallow: /translate suggestion

Disallow: /profiles/me  
Allow: /profiles  
Disallow: /s2/profiles/me  
Allow: /s2/profiles  
Allow: /s2/photos  
Allow: /s2/static  
Disallow: /s2  
Disallow: /transconsole/portal/  
Disallow: /gcc/  
Disallow: /aclk  
Disallow: /cse?  
Disallow: /cse/panel  
Disallow: /cse/manage  
Disallow: /tbproxy/  
Disallow: /comparisonads/  
Disallow: /imesync/  
Disallow: /shenghuo/search?  
Disallow: /support/forum/search?  
Disallow: /reviews/polls/  
Disallow: /hosted/images/  
Disallow: /hosted/life/  
Disallow: /ppob/?  
Disallow: /ppob?  
Disallow: /ig/add?  
Disallow: /adwordsresellers  
Disallow: /accounts/o8  
Allow: /accounts/o8/id  
Disallow: /topicsearch?q=  
Disallow: /xfx7/  
Disallow: /squared/api  
Disallow: /squared/search  
Disallow: /squared/table  
Disallow: /toolkit/  
Allow: /toolkit/\*.html  
Disallow: /qnasearch?  
Disallow: /errors/  
Disallow: /app/updates  
Disallow: /sidewiki/entry/  
Disallow: /quality\_form?  
Disallow: /labs/popgadget/search  
Disallow: /buzz/post  
Sitemap: http://www.gstatic.com/s2/sitemaps/profiles-sitemap.xml  
Sitemap: http://www.google.com/hostednews/sitemap\_index.xml  
Sitemap: http://www.google.com/ventures/sitemap\_ventures.xml  
Sitemap: http://www.google.com/sitemaps\_webmasters.xml  
Sitemap: http://www.gstatic.com/trends/websites/sitemaps/sitemapindex.xml  
Sitemap: http://www.gstatic.com/dictionary/static/sitemaps/sitemap\_index.xml

10.1.1. Sitemap in robots.txt

User-agent: \*  
Allow: \*  
Disallow: /management/  
Sitemap: http://netkiller.sourceforge.net/sitemaps.xml.gz

10.2. sitemaps

<http://www.sitemaps.org/>

sitemap.xml





# 第 13 章 Application Design

## 目录

- [1. Product](#)
- [2. Cart & Checkout](#)
- [3. Other](#)

## 1. Product

```
class Product():
    attribute = []
    def __init__(self, name = None):
        if name :
            self.name = name
        else:
            self.name = "Unknown"
            self.description = "None"
            self.price = 0
    def getName(self):
        return self.name
    def getDescription(self):
        return self.description

class Attribute():
    def __init__(self,product, attr):
        self.product = product
        self.product.attribute.append(attr)

class Color():
    def __init__(self):
        pass
    def Red(self):
        return {'color': 'red'}
    def Blue(self):
        return {'color': 'blue'}

class Size():
    def __init__(self):
        pass
    def Small(self):
        return {'size': 'small'}
    def Big(self):
        return {'site': 'big'}
```



## 2. Cart & Checkout

```
class Cart:
    def __init__(self):
        self.products = []
    def add(self, obj):
        self.products.append(obj)
    def notify(self):
        for obj in self.products:
            obj.notify(len(self.products))
    def price(self):
        for obj in self.products:
            print(obj.price)

class Checkout():
    def __init__(self, cart):
        self.cart = cart
        self.products = self.cart.products
        self.totals = 0
        self.accounting()
    def total(self):
        print(self.totals)
        return (self.totals)
    def accounting(self):
        for obj in self.products:
            self.totals = self.totals + obj.price
    def bill(self):
        pass

class Shipping():
    def __init__(self, checkout, ship):
        self.ship = ship
        checkout.totals = checkout.totals + ship.getCost()
    def cost(self):
        print(self.ship.getCost())

class UPS():
    def __init__(self):
        self.cost = 15.2
    def getCost(self):
        return self.cost

class FedEx(Shipping):
    def __init__(self):
        self.cost = 10
    def getCost(self):
        return self.cost

class Payment():
    def __init__(self, checkout):
        pass
    def payable(self):
        pass
    def tax(self):
        pass
```



### 3. Other



# 部分 III. Operations

目录

[14. How Opensource Helps Optimize IT and Save You Money](#)

[15. 运维管理管理](#)

[1. 合同管理](#)

[2. IDC 设备管理](#)

[2.1. 资产管理](#)

[2.2. 图表管理](#)

[2.2.1. 网络拓扑图](#)

[2.2.2. 机柜图](#)

[2.2.3. 防火墙/路由器与交换机链路图](#)

[2.2.4. 服务器连接图](#)

[2.3. 进出IDC机房](#)

[2.4. 设备上下架](#)

[2.5. 如何维护设备](#)

[16. IDC](#)

[1. 接入线路](#)

[1.1. 带宽计算](#)

[1.2. 双线/多线](#)

[1.2.1. 智能DNS \(Smart DNS\)](#)

[1.3. BGP](#)

[1.4. 接入线路测试](#)

[2. 机柜\(Cabinets\)](#)

[2.1. 强电相关](#)

[2.2. IP 分配](#)

[3. Network](#)

[3.1. Firewall](#)

[3.1.1. DIY Firewall & VPN](#)

[3.1.1.1. Firewall](#)

[3.1.1.2. 3 Layer VPN](#)

[3.1.1.3. 7 Layer VPN](#)

[3.2. Switch](#)

[3.3. 物理拓扑图](#)

[4. 服务器部署与网络拓扑](#)

[4.1. 小型网站](#)

[4.2. 集群网站](#)

[4.3. 聚合端口](#)

[4.4. 关于服务器远程管理](#)

[5. RALDRS](#)

[17. Hardware](#)

[1. Server](#)

[1.1. TPC](#)

[1.2. IOPS \(Input/Output Operations Per Second, pronounced i-ops\)](#)

[1.3. rPerf](#)

[1.4. 网卡](#)

[1.4.1. 网口规划](#)

[1.4.1.1. 内外隔离](#)

[1.4.1.2. 负载均衡](#)

[1.4.1.3. 交叉互联](#)

[1.4.2. 常见网络适配器品牌](#)

[1.4.3. 1G 千兆以太网产品](#)

[1.4.4. 10G 万兆以太网产品](#)

[1.5. Fibre Channel HBA 卡](#)

[1.6. iDRAC / iLO / IMM](#)

[1.7. Hardware configuration](#)

[1.7.1. www/app](#)

[1.7.2. cache](#)

[1.7.3. database](#)

[1.8. 密码管理](#)

## [2. Network](#)

[2.1. Firewall](#)

[2.1.1. Cisco ASA](#)

[2.1.2. Juniper](#)

[2.2. Switch](#)

[2.2.1. 2960G / 2960S](#)

[2.2.2. 3560 / 3750](#)

[2.2.3. 4948](#)

[2.2.4. 4506 / 4507](#)

[2.3. SLB](#)

## [18. Linux Server](#)

[1. 服务器快速安装与部署](#)

[1.1. Kickstart](#)

[1.2. 快速初始化](#)

[1.3. JDK and Tomcat](#)

[1.4. 批量安装](#)

[2. Kernel](#)

[2.1. RPS / RFS](#)

[3. IP / Mac Address](#)

[3.1. IP Address](#)

[3.2. Mac Address](#)

[3.3. NIC](#)

[3.4. DNS](#)

[3.5. bonding](#)

[4. limits.conf](#)

[5. sysctl](#)

[6. profile](#)

[7. Crontab](#)

[8. NFS](#)

[9. 释放内存](#)

[10. OpenSSH](#)

[10.1. history 安全](#)

[10.2. sshd\\_config](#)

[11. TCP/IP](#)

[11.1. TCP 状态](#)

[11.2. TIME\\_WAIT](#)

[11.3. SYN\\_RECV / SYN\\_SENT](#)

[12. NTPD](#)

[13. dnsmasq](#)

[14. snmp](#)

[15. nrpe](#)

[16. vsftpd](#)

[17. 安装 LAMP](#)

[17.1. Apache](#)

[17.2. MySQL](#)

[17.3. PHP](#)

[17.4. resin-pro](#)

[17.5. httpd](#)

[17.6. php](#)

[17.7. httpd-2.2.21 + php-5.3.8 自动安装脚本](#)

[18. Linux 日常管理](#)

[20. Monitor solution](#)

[1. 网络监控](#)

[1.1. 流量监控](#)

[1.2. 交换机监控](#)

[2. 集群监控](#)

[3. 服务器状态监控](#)

[3.1. 负载监控](#)

[3.2. 磁盘空间监控](#)

[3.3. 内存监控](#)

[3.4. CPU监控](#)

[4. MySQL 监控](#)

[5. 网站安全与预警机制](#)

[21. Backup](#)

[1. help](#)

[1.1. Task](#)

[1.2. Schedule](#)

[1.3. Crontab](#)

[2. 配置文件备份](#)

[2.1. Firewall and Switch](#)

[2.2. Server](#)

[22. 常用工具](#)

[1. IP Zone](#)

[2. What is my IP address?](#)

[2.1.](#)

[3. IP Tracking](#)

[4. 自动化脚本](#)

[4.1. Install Script](#)

[4.2. FTP User Script](#)

[4.3. SVN User Script](#)

[23. 服务器安全](#)

[1. 防火墙](#)

[2. Web Server](#)

[2.1. 目录访问控制](#)

[2.2. fastcgi 安全](#)

[2.3. php 函数安全](#)

[2.4. 服务器版本信息](#)

[24. 运维自动化](#)

[1. Cobbler](#)

[2. Cfengine](#)



[3. Puppet](#)

[4. func](#)

[上一页](#)

[下一页](#)

3. Other

[起始页](#)

第 14 章 How Opensource Helps Optimize IT  
and Save You Money

[Home](#) | [Mirror](#) | [Search](#)



# 第 14 章 How Opensource Helps Optimize IT and Save You Money

OS Linux/FreeBSD

Web Apache/Lighttpd/Nginx

DB MySQL/PostgreSQL



# 第 15 章 运维管理管理

## 目录

### [1. 合同管理](#)

### [2.IDC 设备管理](#)

#### [2.1. 资产管理](#)

#### [2.2. 图表管理](#)

##### [2.2.1. 网络拓扑图](#)

##### [2.2.2. 机柜图](#)

##### [2.2.3. 防火墙/路由器与交换机链路图](#)

##### [2.2.4. 服务器连接图](#)

#### [2.3. 进出IDC机房](#)

#### [2.4. 设备上下架](#)

#### [2.5. 如何维护设备](#)

## 1. 合同管理



## 2. IDC 设备管理

### 2.1. 资产管理

设备登记

### 2.2. 图表管理

2.2.1. 网络拓扑图

2.2.2. 机柜图

2.2.3. 防火墙/路由器与交换机链路图

接口对应关系

交换机接口对应服务器的关系

2.2.4. 服务器连接图

server A ----> B server 服务器相互访问关系

server:80 A ----> D server:80 端口及服务之间的关系

### 2.3. 进出IDC机房

进出IDC机房注意事项

### 2.4. 设备上下架

上下架流程

### 2.5. 如何维护设备

[Home](#) | [Mirror](#) | [Search](#)



# 第 16 章 IDC

## 目录

### [1. 接入线路](#)

[1.1. 带宽计算](#)

[1.2. 双线/多线](#)

[1.2.1. 智能DNS \(Smart DNS\)](#)

[1.3. BGP](#)

[1.4. 接入线路测试](#)

### [2. 机柜\(Cabinets\)](#)

[2.1. 强电相关](#)

[2.2. IP 分配](#)

### [3. Network](#)

[3.1. Firewall](#)

[3.1.1. DIY Firewall & VPN](#)

[3.1.1.1. Firewall](#)

[3.1.1.2. 3 Layer VPN](#)

[3.1.1.3. 7 Layer VPN](#)

[3.2. Switch](#)

[3.3. 物理拓扑图](#)

### [4. 服务器部署与网络拓扑](#)

[4.1. 小型网站](#)

[4.2. 集群网站](#)

[4.3. 聚合端口](#)

[4.4. 关于服务器远程管理](#)

## 1. 接入线路

### 1.1. 带宽计算

### 1.2. 双线/多线

#### 1.2.1. 智能DNS (Smart DNS)

### 1.3. BGP

### 1.4. 接入线路测试

用于评估机房线路.

ping.lst

```
202.96.134.133
202.96.128.68
...
...
```

测试脚本

```
#!/bin/bash
for addr in `cat ping.lst`
do
    ping -c 5 -s 128 -W 5 $addr
    echo
done > ping_report.txt &
```

将不同机房的出ping\_report.txt文件汇总，然后用是diff对比。例如 vimdiff report1.txt report2.txt.

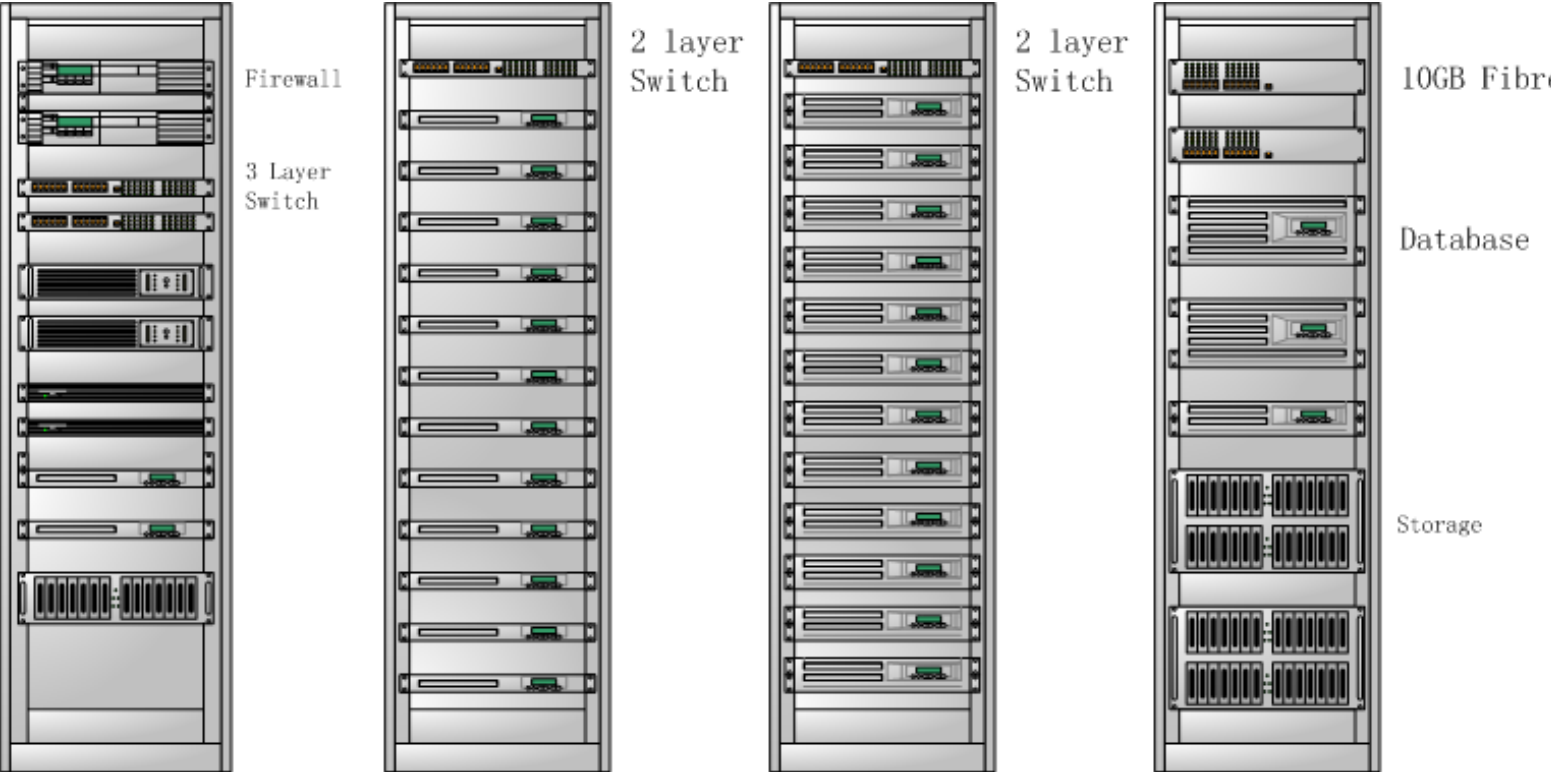
```
#!/bin/bash
for addr in `cat ping.lst`
do
    traceroute $addr
    echo
done > traceroute_report.txt &
```

## 2. 机柜(Cabinets)

目前比较新的五星级机房，前门都是全铁门，后门是网面双开门，老IDC很多是玻璃门。空调系统是机柜摆放背对背，然后从前门地下向上送风，后门过道抽走热风。

机柜前面放入服务器，每个服务器一个隔板，服务器之间要有1U的空间，便于空气流通。

打开机柜后门，左侧是网线槽，右侧是强电槽



### 2.1. 强电相关

一般是每个机柜分配15A（安培）的电力供应，也有一些IDC是规定13A。据说花钱可以增加。

右侧壁上会提供10组20个电源插座; 每个插孔均配有空气开关，这样可以防止单个服务器短路，导致整个机柜掉电。

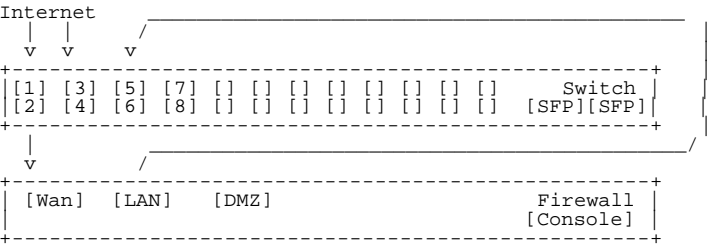
一般42U的机柜最多能够放置15台1U服务器加一个交换机。有些企业被忽悠买了刀片服务器，发现一个机柜只能放置一个刀笼（因为电力不够，只能提供15安培）。

### 2.2. IP 分配

每个机柜一般只能放15台服务器，可以给每个机柜分配20 或 40 个IP地址

### One Firewall + One 3Layer Switch

```
=====
Design by Neo chan
Homepage: http://netkiller.sf.net
=====
```



```
Vlan 2
-----
G0/0/1-4

Port
-----
G0/0/1      Internet Active Line
G0/0/3      Internet Stand by Line
G0/0/4      Debug, Mirror or Other

G0/0/2      Firewall Outside
G0/0/5      Firewall Inside

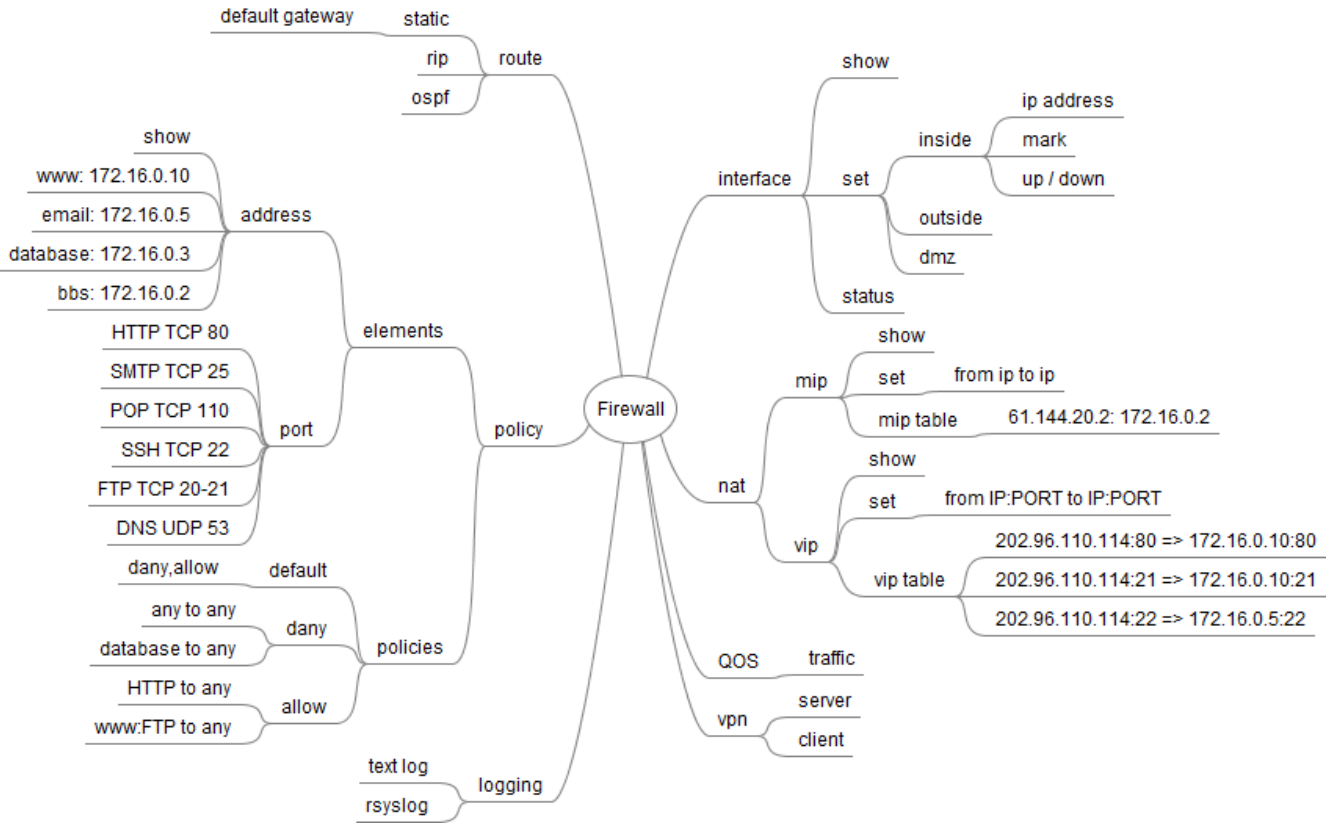
G0/0/6-24   Server
```

### 3.1.1. DIY Firewall & VPN

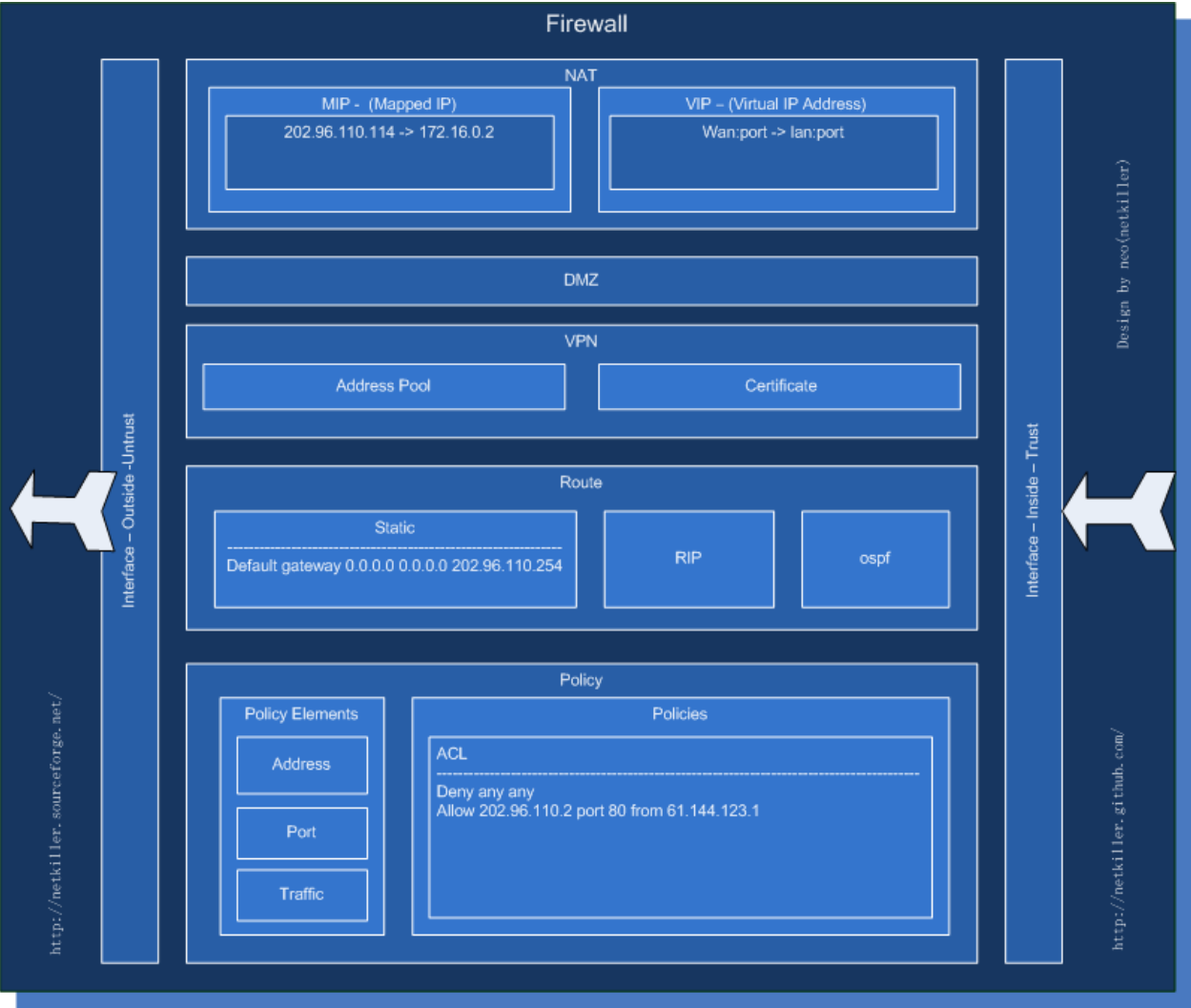
If you do not have enough money to buy a firewall, I'll tell you how to do a one.

Iptables + TC + Openvpn / PPTP | L2TP

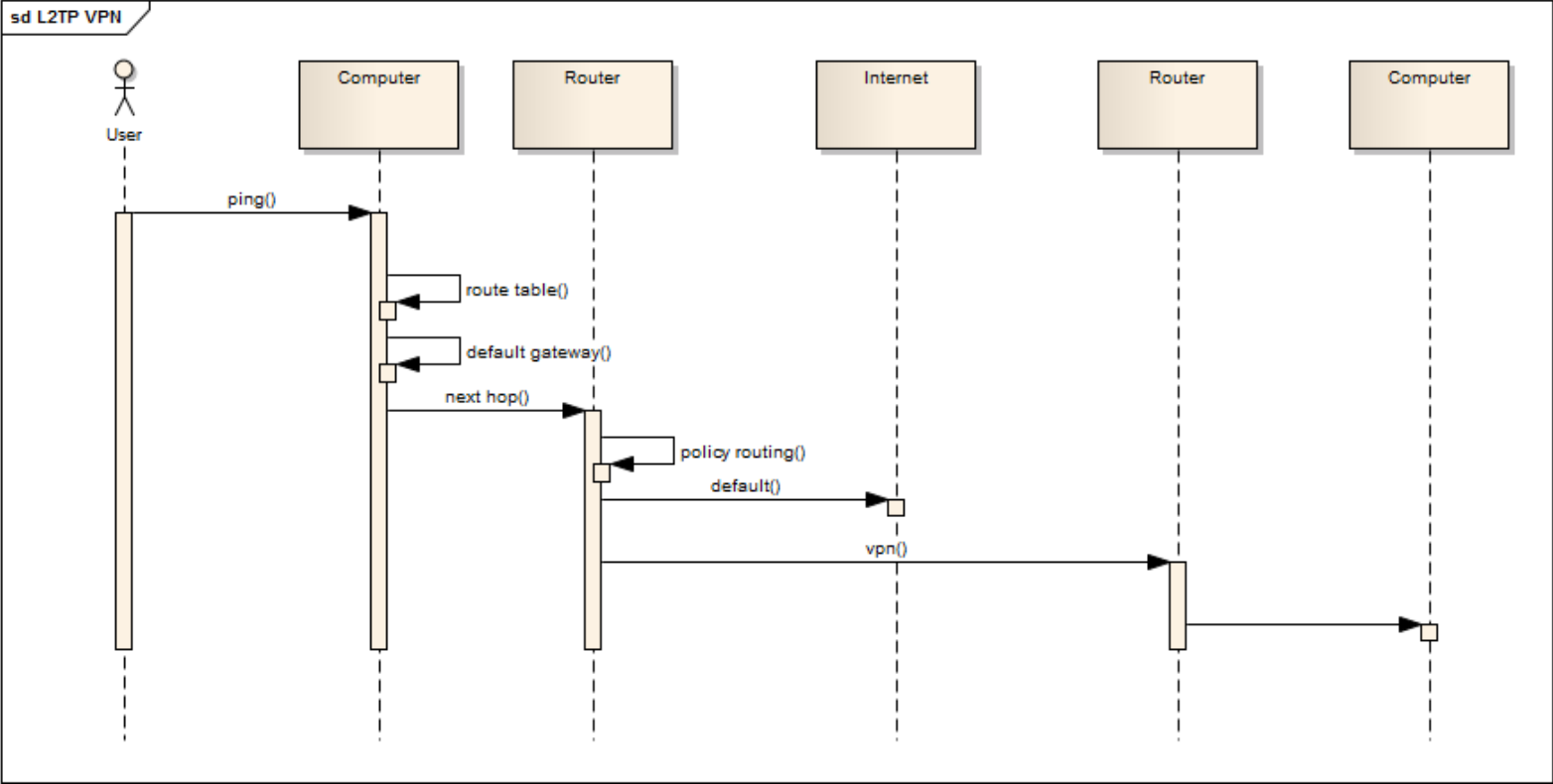
#### 3.1.1.1. Firewall



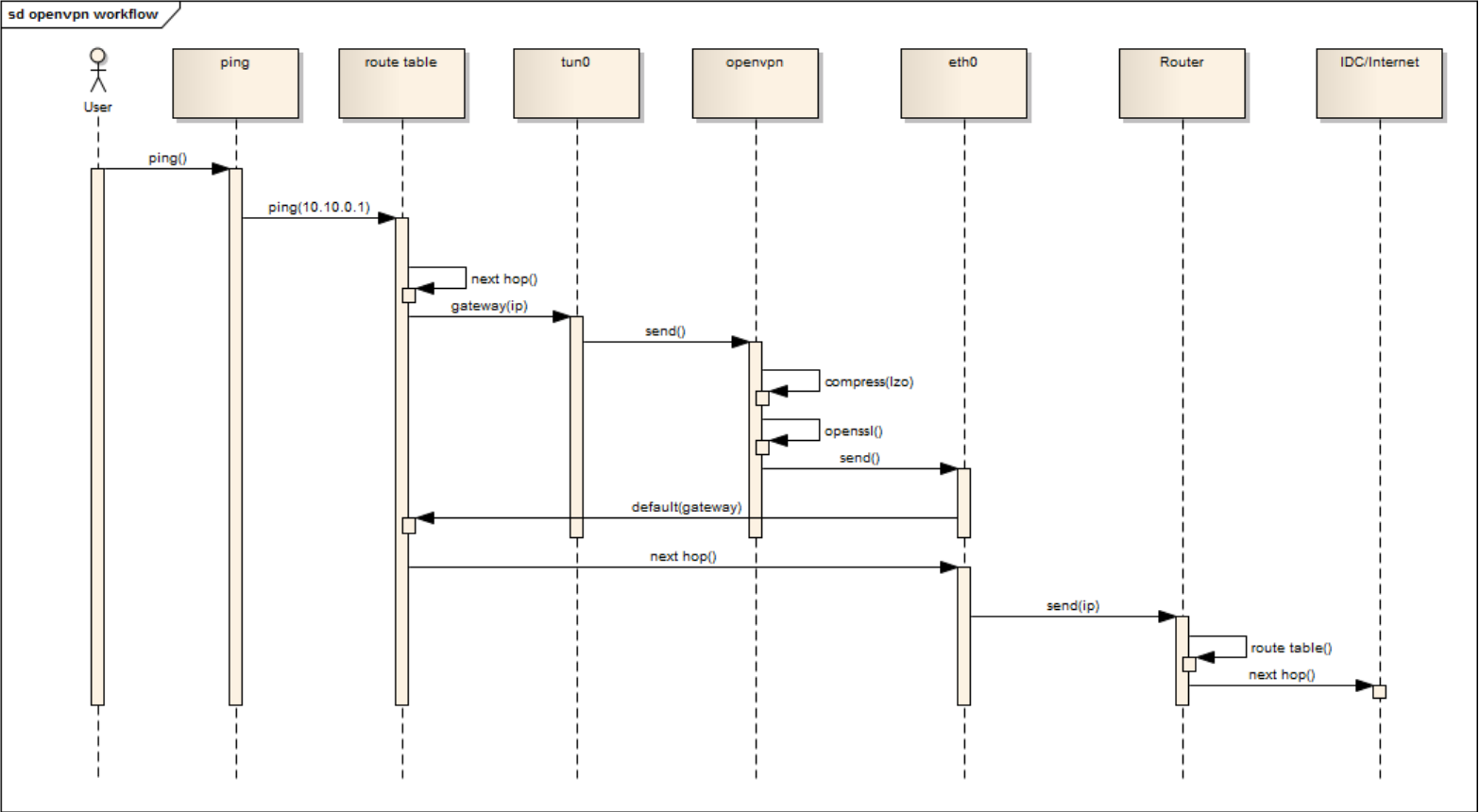




3.1.1.2. 3 Layer VPN

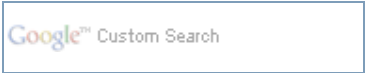


3.1.1.3. 7 Layer VPN



3.2. Switch

3.3. 物理拓扑图



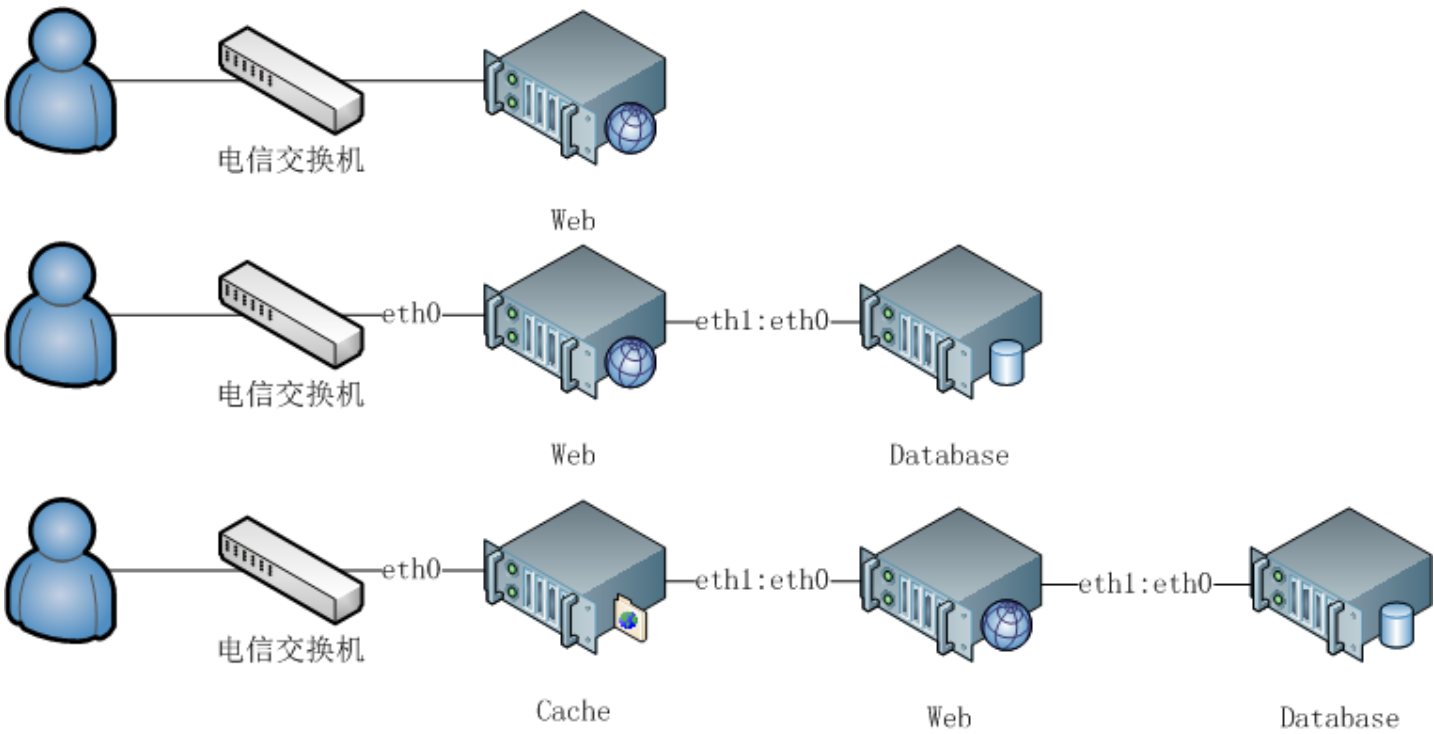
## 4. 服务器部署与网络拓扑

所谓大型网站主要的特点是访问量大，既海量访问，对带宽要求大，而在中国的网络环境比较复杂，单单靠一家也难保访问的快速、稳定，我们可能选择多家网络运营商，才能得到保障。

我们不得不把服务器分散部署到各地

### 4.1. 小型网站

一个IP，多台服务器流水线方式解决方案。



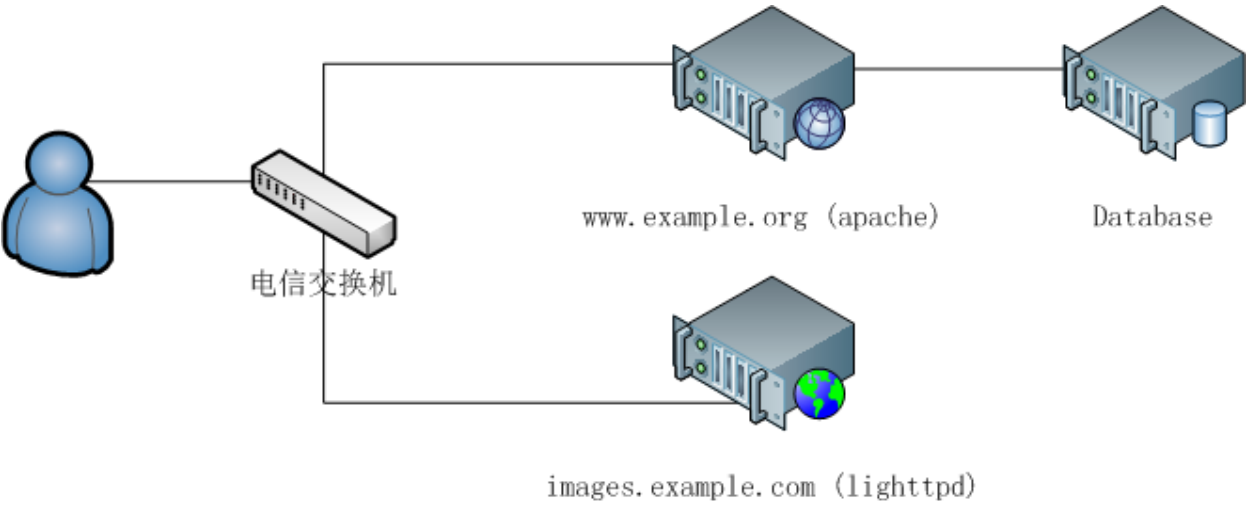
这种方案必须使用带有双网卡的服务器，建议选择千兆网卡，web服务器与database连接建议采用交叉线互联，不要通过Hub,Switch连接。

当web和database在同一台服务器是建议采用 UNIX SOCK 来链接数据库，以代替TCP/IP Socket。

以上图中的第三套方案为例，当用户访问网站时，通过电信交换机连接到cache服务器，有两种情况。第一种是静态html文件或图片，将判断是否被缓存，如果缓存直接反馈给用户否则链接web服务器。第二种动态脚本，将处理立即送给web服务器。如果动态脚本有请求数据库操作，将连接database服务器。

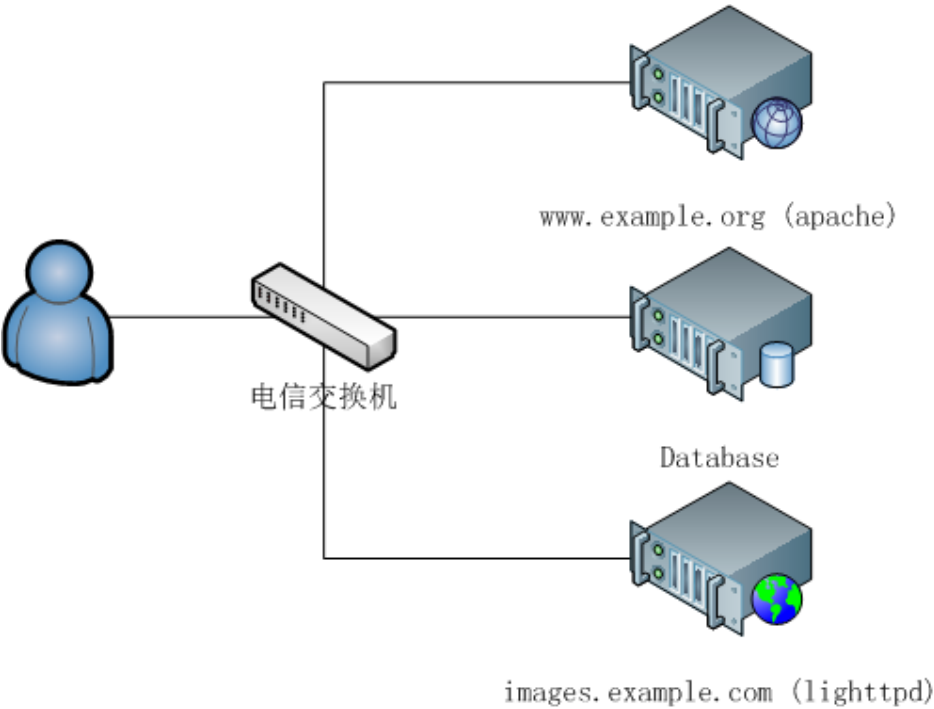
这种方式适合中小型企业，非互联网运营商，仅仅是用一个IP实现。

两个IP，多台服务器解决方案。



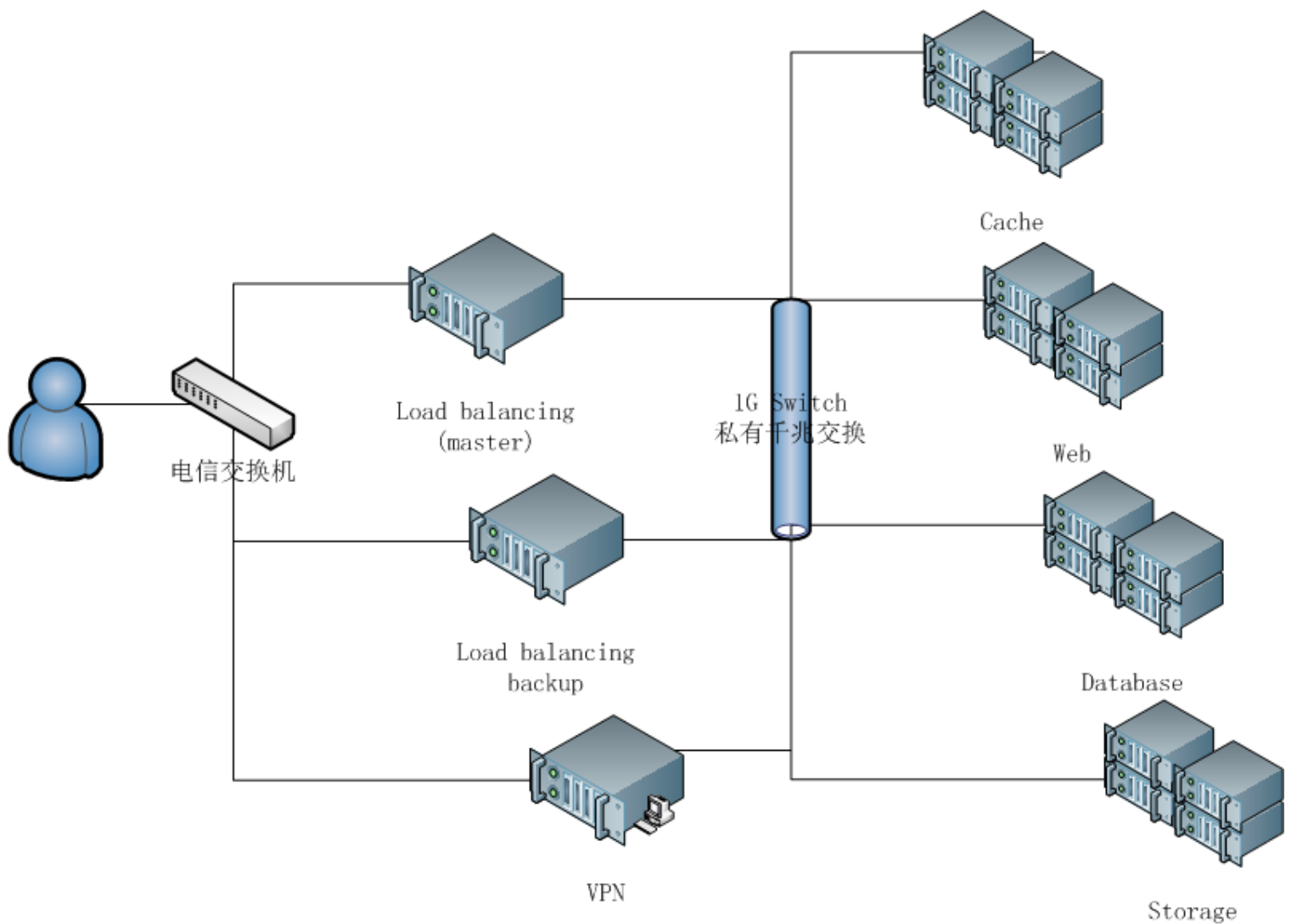
建议你吧图片，缩图单独使用一台服务器实现。

多个IP，多台服务器解决方案。



这种方案要注意服务器全部暴露在WAN上，一定要谨慎设置iptables规则。

#### 4.2. 集群网站



下面是负载均衡的例子

#### 过程 16.1. 过程访问演示

1. 访问用户 `www.example.com` 网站

用户输入网址: `www.example.com`回车

2. dns

域名服务器将`www.example.com`解析到load balancing负载请均衡调度服务器。

3. load balancing

负载均衡器根据调度算法分配到某个squid节点上

4. squid

静态规则：判段是否是缓存，如果已经缓存从cache中直接取出内容，否则请求web服务器

动态规则：直接请求web服务器

请求web服务器是将再次由load balancing分配web节点

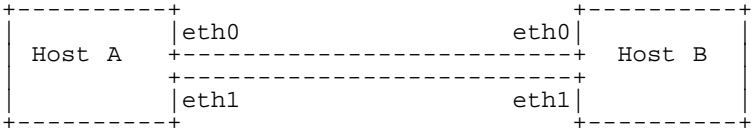
5. web

web server 处理动态脚本，连接数据库查询。连接数据库是仍然需要由load balancing分配database节点

6. database

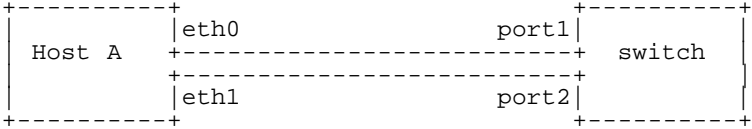
4.3. 聚合端口

Example 1 : host to host at double speed



```
On each host :
# modprobe bonding miimon=100
# ifconfig bond0 addr
# ifenslave bond0 eth0 eth1
```

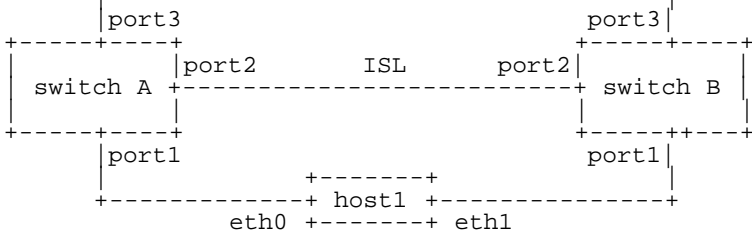
Example 2 : host to switch at double speed



```
On host A :
# modprobe bonding miimon=100
# ifconfig bond0 addr
# ifenslave bond0 eth0 eth1

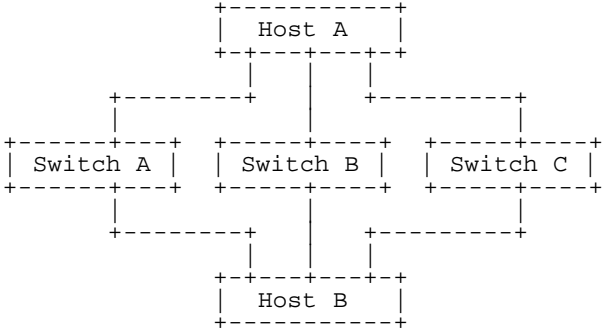
On the switch :
# set up a trunk on port1
and port2
```

Example 3: High Availability in a Multiple Switch Topology

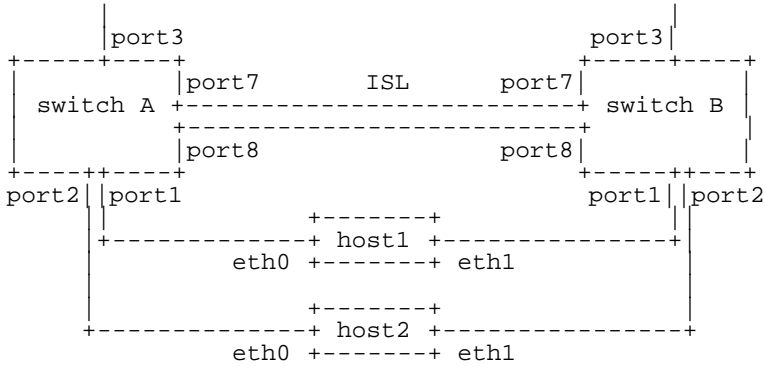


Example 4: Maximum Throughput in a Multiple Switch Topology

Multiple switches may be utilized to optimize for throughput when they are configured in parallel as part of an isolated network between two or more systems, for example:



Example 5: Using multiple host and multiple switches to build a "no single point of failure" solution.



4.4. 关于服务器远程管理

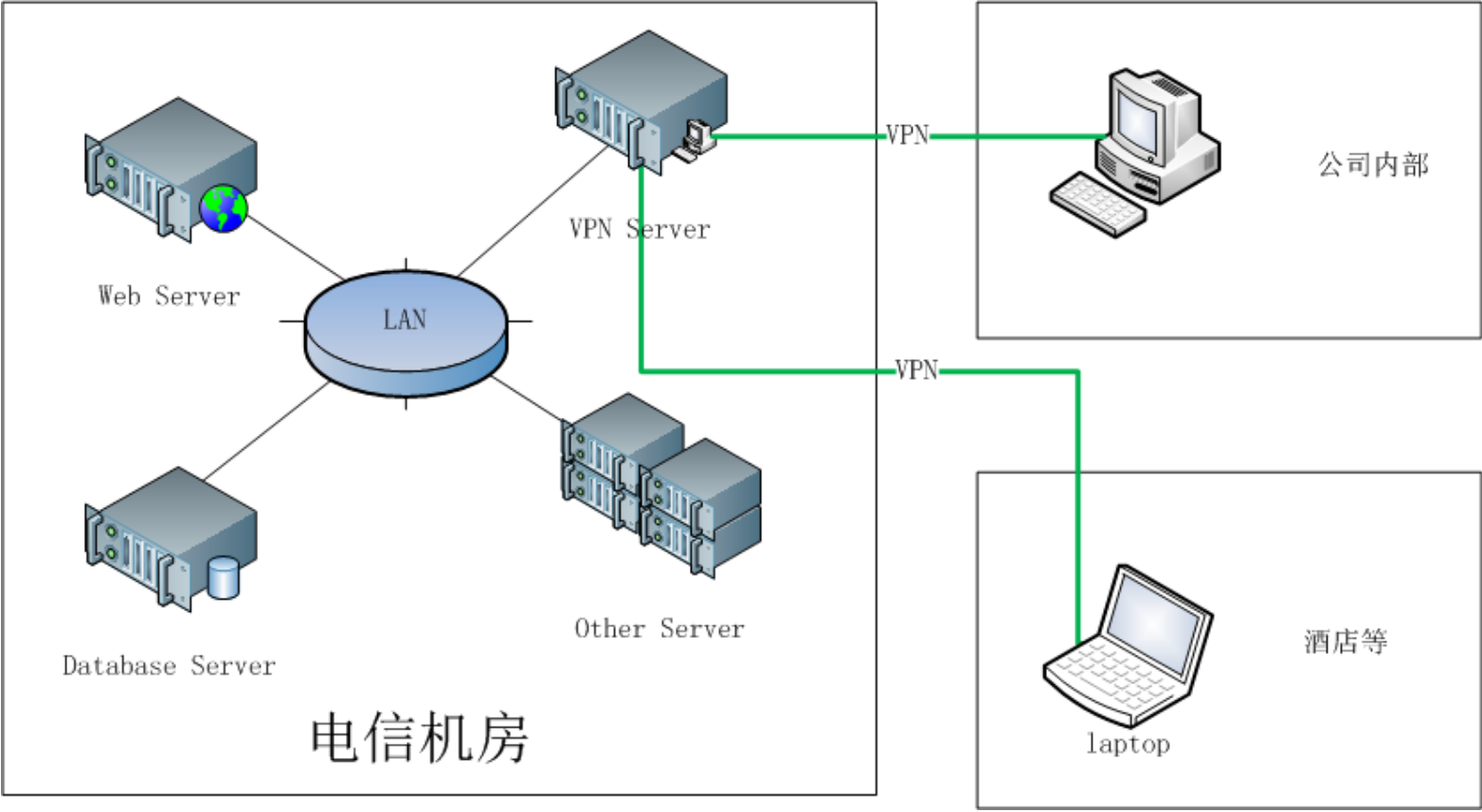
将每一台服务器暴露在广域网上不是一个好主意，我的经验是，目前服务器基本都是双网卡。

eth0: 用于对外访问，如web服务器仅仅开放80端口，其他端口不允许在广域网上直接访问。

eth1: 对内有一个私有局域网，开放SSH的22端口。22端口仅仅能通过私有局域网访问，不能通过广域网访问。

在机房放置一台专门的管理服务器并安装有VPN服务，当远程管理服务器是通过VPN连接到这台服务器，登录到这台服务器上，然后在通过SSH登录到私有LAN上的其他服务器进行远程管理。

下面是VPN解决方案图





## 5. RALDRS

remote application layer disaster recovery system(RALDRS)





# 第 17 章 Hardware

## 目录

### [1. Server](#)

#### [1.1. TPC](#)

#### [1.2. IOPS \(Input/Output Operations Per Second, pronounced i-ops\)](#)

#### [1.3. rPerf](#)

#### [1.4. 网卡](#)

##### [1.4.1. 网口规划](#)

###### [1.4.1.1. 内外隔离](#)

###### [1.4.1.2. 负载均衡](#)

###### [1.4.1.3. 交叉互联](#)

##### [1.4.2. 常见网络适配器品牌](#)

##### [1.4.3. 1G 千兆以太网产品](#)

##### [1.4.4. 10G 万兆以太网产品](#)

#### [1.5. Fibre Channel HBA 卡](#)

#### [1.6. iDRAC / iLO / IMM](#)

#### [1.7. Hardware configuration](#)

##### [1.7.1. www/app](#)

##### [1.7.2. cache](#)

##### [1.7.3. database](#)

#### [1.8. 密码管理](#)

### [2. Network](#)

#### [2.1. Firewall](#)

##### [2.1.1. Cisco ASA](#)

##### [2.1.2. Juniper](#)

2.2. Switch

[2.2.1. 2960G / 2960S](#)

[2.2.2. 3560 / 3750](#)

[2.2.3. 4948](#)

[2.2.4. 4506 / 4507](#)

[2.3. SLB](#)

1. Server

1.1. TPC

<http://www.tpc.org/>

Transaction Processing Performance Council

- 1. TPC-C: 是在线事务处理(OLTP)的基准程序
- 2. TPC-D: 是决策支持(Decision Support) 的基准程序
- 3. TPC-E: 作为大型企业(Enterprise)信息服务的基准程序
- 4. TPC-H: DecisionSupportforAdHocQueries基于特定查询的决策支持
- 5. TPC-W: Webe-Commerce (互联网及电子商务)
- 6. TPC-R: DecisionSupportforBusinessReporting (基于商业报告的决策支持)

1.2. IOPS (Input/Output Operations Per Second, pronounced i-ops)

<http://www.storageperformance.org/home/>

1.3. rPerf

<http://www-03.ibm.com/systems/power/hardware/notices/rperf.html>

服务器所需要的rPerf 值=SUM(NU \* TX \* CS/PP) / MC

NU: 高峰时并发的用户数

TX: 高峰时每个用户的交易数量

CS: 在rPerf=1的服务器上，每个交易所需要的CPU秒

PP: 高峰持续的时间

MC: 最大的CPU利用率 (推荐< 70%)

下面举例说明如何计算所需的rPerf 值，假定某公司的情况如下：

业务高峰时间： 10：00-11：00=1Hour=3600秒

交易类型： 无复杂查询的简单应用

相对交易类型，用户数目分布：轻的=2000， 一般=50， 重的=5

在高峰时，每个用户的交易数量：

- 轻的=120交易/用户
- 一般=60交易/用户
- 重的=15交易/用户

对于rPerf=1的服务器，每个交易响应的CPU秒

- 轻的=1
- 一般=3
- 重的=15

最大的CPU利用率：60%

根据上述公式，可推算出不同交易类型所对应的rPerf值。

轻的交易：NU\*TX\*CS/PP=2000\*120\*1/3600=66.0

一般交易：NU\*TX\*CS/PP=50\*60\*3/3600=2.5

重的交易：NU\*TX\*CS/PP=5\*15\*15/3600=0.3

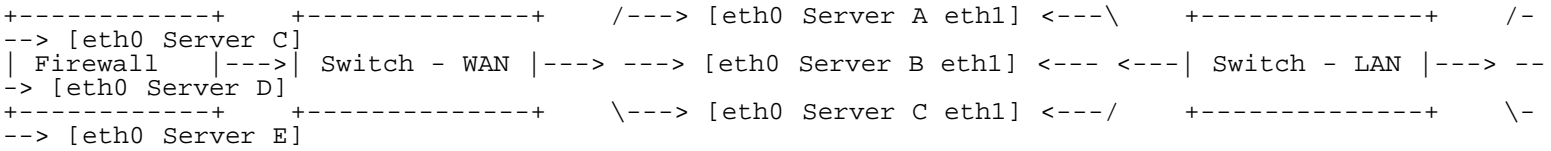
所需的总的rPerf/MC=(66.0+2.5+0.3)/0.7=98.3 rPerf

## 1.4. 网卡

### 1.4.1. 网口规划

#### 1.4.1.1. 内外隔离

双网卡方案,一般服务器会提供至少2块网卡。 使用两个交换机，一个交换机连接接防火墙，另一个交换机独立不接入Internet



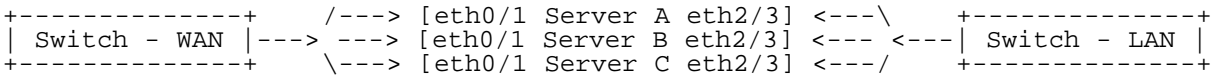
Internet 用户从防火墙进入，只能访问WAN交换机上的服务器，WAN上一般是WEB服务器，WEB服务器通过LAN交换机访问数据库，Memcache等服务器

这样既有效利用了网络IO，有能有效隔离不需要暴露在公网上的服务器还可以降低成本， WAN 可以使用100M交换机，LAN 可以使用1G交换机，因为内部数据传输远远大于外部。

另外 WAN与LAN也可以使用VLAN实现

#### 1.4.1.2. 负载均衡

eth0与eth1 做bonding, eth2与3做bonding, 然后内外隔离



#### 1.4.1.3. 交叉互联

在交换机端口有限的情况可以采用交叉互联。

交叉线连接与通过交换机连接二者差异：

A与B两个服务器举例：

- 交叉线连接A与B两个服务器，A发数据包，B接收数据包，如果接受方在接收包过程中出现异常（毫秒级），可能会堵塞，数据包会重新发包。交换机存储转发，仍然会接收数据放到背板缓存中，建立连接后交换机会处理一切。
- A 服务器出现故障宕机，A网卡灯不亮，那么B服务器的网卡将检测，认为没有插网线，B网卡灯也是不亮状态。而通过交换机B网卡仍然工作

1.4.2. 常见网络适配器品牌

Broadcom NetXtreme II Gigabit Ethernet Driver bnx2 v2.0.8-rh (Oct 11, 2010)

Emulex OneConnect 10Gbps NIC

Intel 10 Gigabit AT2 Server Adapter (E10G41AT2)

1.4.3. 1G 千兆以太网产品

目前服务器1G网卡市场90%都被Broadcom NetXtreme占领，不仅仅限于服务器网卡，Cisco的设备中用的也是Broadcom NetXtreme芯片

1.4.4. 10G 万兆以太网产品

万兆以太网标准很多，有10000BAST-T(使用双绞线连接)，还有SFP+(850nm 光纤连接)

Dell 有通过6类线连接的万兆交换机8024，服务器端Dell给用户配的是Intel万兆网卡，使用方法与千兆一样。

笔者有两个刀笼（刀片服务器），刀笼配置万兆模块通过4条10G SFP+ 连接到8024，然后服务器使用6类双绞线，通过Intel网卡连接8024。

光纤万兆网卡与千兆网卡使用上并无不同。如果指示灯不亮，请调换RX/TX光纤跳线

```
# dmesg | grep Emulex
Copyright(c) 2004-2009 Emulex. All rights reserved.
be2net 0000:18:00.0: eth0 - Emulex OneConnect 10Gbps NIC
be2net 0000:18:00.1: eth1 - Emulex OneConnect 10Gbps NIC
```

笔者使用过Emulex/Intel在Linux上无需驱动，光纤交换机Cisco 4507的万兆模块是Broadcom NetXtreme芯片的.

提示

无论是外形还是接口，万兆以太网与FC(Fibre Channel) HBA 卡很难区分，且卡上没有任何印刷文字提示，购买千万小心不要买错，最好与厂商反复确认。另外光纤交换机与FC交换机也容易混淆，我建议你网卡用Cisco交换机，存储用博科交换机

1.5. Fibre Channel HBA 卡

常用HBA卡

QLogic QLE2562 - PCI-Express Dual Channel 8Gb Fibre Channel HBA

Emulex LightPulse Fibre Channel SCSI driver 8.2.0.87.1p

HBA 卡使用SFP+光纤模块，LC-LC光纤跳线

1.6. iDRAC / iLO / IMM

远程管理卡

1.7. Hardware configuration

1.7.1. www/app

Processor: 5650

Memory: 32GB

Maximum Internal Storage: 146GB\*4, 2.5" drives

/boot 200M  
/ 30G  
/var 50G  
Swap mem \* 2  
/www

1.7.2. cache

Processor: 7500

Memory: 64GB

Maximum Internal Storage: 146GB\*2, 2.5" drives

1.7.3. database

Processor: 7500

Memory: 64GB

Maximum Internal Storage: 146GB\*8, 2.5" drives

Maximum Storage: 2TB\*12, 3.5" drives

1.8. 密码管理

[KeePass Password Safe](#)





## 2. Network

### 2.1. Firewall

#### 2.1.1. Cisco ASA

ASA 5550 会话数 65000

购买是需要注明BUN-K9版本的IOS

#### 2.1.2. Juniper

### 2.2. Switch

#### 注意事项

Cisco 与 H3C 的交换机软件都有简版与企业版之分，购买一定要注意，尤其是买三层交换机时。Juniper我的交换机我没有用过。

例如我曾经购买H3C的三层交换机SI版，竟然没有DHCP功能，后来找厂家升级到EI版

例如 3750 简版OSPF。如果你想使用OSPF需要企业版

例如 4507 简版没有策略路由功能，需要企业版才支持。

#### 2.2.1. 2960G / 2960S

#### 2.2.2. 3560 / 3750

#### 2.2.3. 4948

#### 2.2.4. 4506 / 4507

购买4507 时要注意引擎版本

### 2.3. SLB





[Home](#) | [Mirror](#) | [Search](#)



# 第 18 章 Linux Server

## 快速初始化Linux服务器

### 目录

#### [1. 服务器快速安装与部署](#)

- [1.1. Kickstart](#)
- [1.2. 快速初始化](#)
- [1.3. JDK and Tomcat](#)
- [1.4. 批量安装](#)

#### [2. Kernel](#)

- [2.1. RPS / RFS](#)

#### [3. IP / Mac Address](#)

- [3.1. IP Address](#)
- [3.2. Mac Address](#)
- [3.3. NIC](#)
- [3.4. DNS](#)
- [3.5. bonding](#)

#### [4. limits.conf](#)

#### [5. sysctl](#)

#### [6. profile](#)

#### [7. Crontab](#)

#### [8. NFS](#)

#### [9. 释放内存](#)

#### [10. OpenSSH](#)

- [10.1. history 安全](#)
- [10.2. sshd\\_config](#)

#### [11. TCP/IP](#)

[11.1. TCP 状态](#)

[11.2. TIME\\_WAIT](#)

[11.3. SYN\\_RECV / SYN\\_SENT](#)

[12. NTPD](#)

[13. dnsmasq](#)

[14. snmp](#)

[15. nrpe](#)

[16. vsftpd](#)

[17. 安装 LAMP](#)

[17.1. Apache](#)

[17.2. MySQL](#)

[17.3. PHP](#)

[17.4. resin-pro](#)

[17.5. httpd](#)

[17.6. php](#)

[17.7. httpd-2.2.21 + php-5.3.8 自动安装脚本](#)

[18. Linux 日常管理](#)

# 1. 服务器快速安装与部署

## 1.1. Kickstart

使用 Kickstart 实现无人值守安装，制作一个Linux 安装光盘或ISO文件将ks文件放置到根目录。这样光盘放入光驱后无需人为干预，实现自动安装。

另外还可以采用无盘启动网络安装

anaconda-ks.cfg

```
# Kickstart file automatically generated by anaconda.

#version=RHEL6
install
cdrom
lang en_US.UTF-8
keyboard us
network --device eth0 --bootproto static --ip 172.16.3.81 --netmask 255.255.255.0 --gateway 172.16.3.254 --nameserver 8.8.8.8 --hostname xop.xiu.com
rootpw --iscrypted $6$Ze/iw9HmY5LVYHhN$Hil8L/e8r2EwzmpacjW7VnZulJx6V9ZE55oAAWNr52qnh82ZL9m9J340mSRQXB5fYI2/ahL09xSlZ2WbX
firewall --service=ssh
authconfig --enableshadow --passalgo=sha512 --enablefingerprint
selinux --enforcing
timezone --utc Asia/Harbin
bootloader --location=mbr --driveorder=sda,sdb --append="crashkernel=auto rhgb quiet"
# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work
#clearpart --none --drives=sda

#part / --fstype=ext4 --grow --size=200
#part swap --size=64000
```

```
#part /www --fstype=ext4 --grow --size=200

repo --name="CentOS" --baseurl=file:///mnt/source --cost=100

%packages
@core
@server-policy
%end
```

## 1.2. 快速初始化

后面章节中提供很多脚本，直接复制下面命令，右键粘贴到终端中运用即可。

例如每个服务器都需要安装ntp,您只需要服务下面代码，粘贴到终端即可，它能实现自动安装，修改配置文件与启动服务等等

```
yum install ntp -y
vim /etc/ntp.conf <<VIM > /dev/null 2>&1
:17,17s/^/server 172.16.1.10\r/
:wq
VIM
service ntpd start
chkconfig ntpd on
```

你也可以吧这些脚本组合，制作一个你自己的脚本，然后批量执行。例如：

```
$ cat install.sh
#!/bin/bash

if [ ! -f /usr/bin/vim ] ; then
    alias vim='vi'
fi

if [ -z "$( egrep "CentOS|Redhat" /etc/issue)" ]; then
    echo 'Only for Redhat or CentOS'
    exit
fi

echo -ne "
search xiu.com
nameserver 172.16.3.51
nameserver 172.16.3.52
nameserver 208.67.222.222
nameserver 202.67.220.220
nameserver 8.8.8.8
nameserver 4.4.4.4
" > /etc/resolv.conf

echo -ne "

* soft nofile 65536
* hard nofile 65536
" >>/etc/security/limits.conf

cat >> /etc/sysctl.conf <<EOF

net.ipv4.ip_local_port_range = 1024 65500
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_fin_timeout = 60
net.ipv4.tcp_keepalive_time = 1200
net.ipv4.tcp_max_syn_backlog = 8192
net.ipv4.tcp_max_tw_buckets = 4096
EOF

yum update -y
rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt
rpm -K http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-2.el6.rf.x86_64.rpm
rpm -i http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-2.el6.rf.x86_64.rpm

# redhat 5.6 cp /etc/ntp.conf.original /etc/ntp.conf
yum install ntp -y
vi /etc/ntp.conf <<VIM > /dev/null 2>&1
:22,24s/^/#/
:25,25s/^/\rserver 172.16.3.51\rserver 172.16.3.52\r/
:wq
VIM
service ntpd start
chkconfig ntpd on

yum install net-snmp -y
vi /etc/snmp/snmpd.conf <<VIM > /dev/null 2>&1
:62,62s/systemview/all/
:85,85s/^#//
:wq
```

```
VIM
service snmpd start
chkconfig snmpd on

yum install -y nrpe nagios-plugins
vi /etc/nagios/nrpe.cfg <<VIM > /dev/null 2>&1
:%s/allowed_hosts=127.0.0.1/allowed_hosts=172.16.1.2/
:wq
VIM

cat >> /etc/nagios/nrpe.cfg <<EOF

#command[check_http]=/usr/lib64/nagios/plugins/check_http -I 127.0.0.1 -p 80 -u
http://www.example.com/index.html
command[check_swap]=/usr/lib64/nagios/plugins/check_swap -w 20% -c 10%
command[check_all_disks]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -e
EOF

chkconfig nrpe on
service nrpe start
```

将上面脚本放到web server 上，然后使用rsh循环批量执行的

```
curl -s http://172.16.0.20/shell/install.sh | bash
```

### 1.3. JDK and Tomcat

```
#!/bin/bash
rsync -auzv root@172.16.3.23:/srv/* /srv/
ln -s /srv/java /usr/local/java
ln -s /srv/apache-tomcat /usr/local/
mkdir -p /www/logs/tomcat
echo -ne '
export JAVA_HOME=/srv/java
export CATALINA_HOME=/srv/apache-tomcat
export CLASSPATH=$JAVA_HOME/lib:$JAVA_HOME/jre/lib:$CATALINA_HOME/lib
export PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$CATALINA_HOME/bin:
export JAVA_OPTS="-server -Xms512m -Xmx8192m -XX:PermSize=64M -XX:MaxPermSize=512m"
' >> /etc/profile
```

### 1.4. 批量安装

检查日期

```
for ip in {23..32} {49,50} {81..92}; do ssh root@172.16.0.$ip date; done
```

时间更新

```
for ip in {23..32} {49,50} {81..92}; do ssh root@172.16.0.$ip ntpdate 172.16.0.1; done
```

安装NFS Server

```
for ip in {21..32} {41,50} {81..92}; do rsh root@172.16.0.$ip "yum install -y nfs-utils"; done
for host in {21..32} {41,50} {81..92}; do echo 172.16.0.$host && rsh root@172.16.0.$host
"exportfs"; done

for ip in {21..32} {41,50} {81..92}; do rsync -auz ~/config/nfs/exports root@172.16.0.$ip:/etc/;
rsh root@172.16.0.$ip "service nfs reload"; done
```



## 2. Kernel

### 2.1. RPS / RFS



### 3. IP / Mac Address

#### 3.1. IP Address

```
nmap -sP 172.16.1.0/24
```

#### 3.2. Mac Address

```
nmap -sP -PI -PT -oN ipandmaclist.txt 172.16.1.0/24
```

#### 3.3. NIC

```
DEVICE=eth0
BOOTPROTO=static
HWADDR=00:15:17:3E:90:90
ONBOOT=yes
NETMASK=255.255.255.0
IPADDR=172.16.1.26
GATEWAY=172.16.1.1
TYPE=Ethernet
USERCTL=no
IPV6INIT=no
PEERDNS=yes
```

#### 3.4. DNS

```
echo -ne "
search example.com
nameserver 208.67.222.222
nameserver 202.67.220.220
nameserver 8.8.8.8
nameserver 4.4.4.4
" > /etc/resolv.conf
```

#### 3.5. bonding

<http://www.kernel.org/pub/linux/kernel/people/marcelo/linux-2.4/Documentation/networking/bonding.txt>

<http://www.linuxfoundation.org/collaborate/workgroups/networking/bonding>

<http://blog.webdir.bg/ubuntu-linux-and-cisco-switch-link-aggregation-load-balancing-bonding/>

```
cat >> /etc/modprobe.d/bonding.conf <<EOF
alias bond0 bonding
options bond0 miimon=100 mode=1 primary=eth0
alias bond1 bonding
options bond1 miimon=100 mode=1 primary=eth2
EOF

cat >> /etc/sysconfig/network-scripts/ifcfg-eth0 <<EOF
DEVICE=eth0
ONBOOT=yes
MASTER=bond0
SLAVE=yes
BOOTPROTO=none
USERCTL=no
```

```
EOF

cat >> /etc/sysconfig/network-scripts/ifcfg-eth1 <<EOF
DEVICE=eth1
ONBOOT=yes
MASTER=bond0
SLAVE=yes
BOOTPROTO=none
USERCTL=no
EOF

cat >> /etc/sysconfig/network-scripts/ifcfg-eth2 <<EOF
DEVICE=eth2
ONBOOT=yes
MASTER=bond1
SLAVE=yes
BOOTPROTO=none
USERCTL=no
EOF

cat >> /etc/sysconfig/network-scripts/ifcfg-eth3 <<EOF
DEVICE=eth3
ONBOOT=yes
MASTER=bond1
SLAVE=yes
BOOTPROTO=none
USERCTL=no
EOF

cat >> /etc/sysconfig/network-scripts/ifcfg-bond0 <<EOF
DEVICE=bond0
ONBOOT=yes
IPADDR=192.168.80.1
NETMASK=255.255.255.0
NETWORK=192.168.80.0
BOOTPROTO=none
USERCTL=no
EOF

cat >> /etc/sysconfig/network-scripts/ifcfg-bond1 <<EOF
DEVICE=bond1
ONBOOT=yes
IPADDR=192.168.80.2
NETMASK=255.255.255.0
NETWORK=192.168.80.0
BOOTPROTO=none
USERCTL=no
EOF
```

```
cp /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth0.old
```

```
cat >> /etc/modprobe.d/bonding.conf <<EOF
alias bond0 bonding
options bond0 mode=balance-alb miimon=1000
EOF

cat > /etc/sysconfig/network-scripts/ifcfg-eth0 <<EOF
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
EOF

cat > /etc/sysconfig/network-scripts/ifcfg-eth1 <<EOF
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
EOF

cat >> /etc/sysconfig/network-scripts/ifcfg-eth2 <<EOF
DEVICE=eth2
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
EOF

cat >> /etc/sysconfig/network-scripts/ifcfg-eth3 <<EOF
DEVICE=eth3
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
EOF

cat >> /etc/sysconfig/network-scripts/ifcfg-bond0 <<EOF
DEVICE=bond0
ONBOOT=yes
BOOTPROTO=none
TYPE=Ethernet
IPADDR=192.168.80.65
NETMASK=255.255.255.0
NETWORK=192.168.80.0
USERCTL=no
EOF
```

```
modprobe bonding mode=balance-alb miimon=1000
ifconfig bond0 up
ifconfig bond0 192.168.80.1 netmask 255.255.255.0 up
```

```
ifenslave bond0 eth0
ifenslave bond0 eth1

echo 1000 > /sys/class/net/bond0/bonding/miimon
echo 6 > /sys/class/net/bond0/bonding/mode
echo balance-alb > /sys/class/net/bond0/bonding/mode

BONDING_OPTS="mode=1 miimon=100"

echo balance-rr >/sys/class/net/bond0/bonding/mode
modprobe bonding mode=balance-tlb miimon=1000

ifenslave -d bond0 eth0 eth1
ifenslave bond1 eth2 eth3

# ethtool eth0
Settings for eth0:
    Supported ports: [ TP ]
    Supported link modes:      10baseT/Half 10baseT/Full
                              100baseT/Half 100baseT/Full
                              1000baseT/Full
    Supports auto-negotiation: Yes
    Advertised link modes:     10baseT/Half 10baseT/Full
                              100baseT/Half 100baseT/Full
                              1000baseT/Full
    Advertised pause frame use: No
    Advertised auto-negotiation: Yes
    Speed: 100Mb/s
    Duplex: Full
    Port: Twisted Pair
    PHYAD: 1
    Transceiver: internal
    Auto-negotiation: on
    MDI-X: Unknown
    Supports Wake-on: g
    Wake-on: d
    Link detected: yes

# more /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.5.0 (November 4, 2008)

Bonding Mode: adaptive load balancing
Primary Slave: None
Currently Active Slave: None
MII Status: down
MII Polling Interval (ms): 1000
Up Delay (ms): 0
Down Delay (ms): 0

[root@r610 ~]# ifconfig bond0 up
[root@r610 ~]# ifenslave bond0 eth0 eth2

# more /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.5.0 (November 4, 2008)

Bonding Mode: adaptive load balancing
Primary Slave: None
Currently Active Slave: eth0
MII Status: up
MII Polling Interval (ms): 1000
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: eth0
MII Status: up
Link Failure Count: 0
Permanent HW addr: 14:fe:b5:dc:30:ee

Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: 14:fe:b5:dc:30:f0

[root@r610 network-scripts]# cat /sys/class/net/bond0/bonding/miimon
1000
[root@r610 network-scripts]# cat /sys/class/net/bonding_masters
bond0
[root@r610 network-scripts]# cat /sys/class/net/bond0/bonding/mode
balance-alb 6
```





## 4. limits.conf

```
ulimit -HSn 65536
ulimit -HSm 65536

echo "ulimit -SHn 65530" >> /root/.bash_profile
echo "ulimit -SHm 16000000" >> /root/.bash_profile
echo "ulimit -HSn 65536" >>/root/.bash_profile

echo -ne "
* soft nofile 65536
* hard nofile 65536
" >>/etc/security/limits.conf > /dev/null 2>&1
```

/etc/security/limits.d

```
cat >> /etc/security/limits.conf <<EOF
root    soft nofile 10240
root    hard nofile 65536
nobody  soft nofile 10240
nobody  hard nofile 65536
EOF
```



## 5. sysctl

```
cat >> /etc/sysctl.conf <<EOF
net.ipv4.ip_local_port_range = 1024 65500
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_keepalive_time = 1200
net.ipv4.tcp_max_syn_backlog = 8192
net.ipv4.tcp_max_tw_buckets = 4096
EOF
/sbin/sysctl -p
```



## 6. profile

```
echo -ne '  
export JAVA_HOME=/usr/local/java  
export RESIN_HOME=/usr/local/resin  
export CLASSPATH=$JAVA_HOME/lib:$JAVA_HOME/jre/lib:$CLASSPATH  
export PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$HOMR/bin:  
export PATH=$PATH:/usr/local/httpd:/usr/local/php:/usr/local/mysql:/usr/local/resin  
export JAVA_OPTS="-server -Xms512m -Xmx4096m -XX:PermSize=64M -XX:MaxPermSize=512m"  
' >> /etc/profile
```



## 7. Crontab

crontab 中输出信息很保存到 /var/spool/clientmqueue 需要定时清除

```
# du -sh /var/* | sort -rn|head -n 10
# cd /var/spool/clientmqueue
# ls | xargs rm -f
```

避免上述情况可以在crontab 中加入 >/dev/null 2>&1

```
# crontab -l
* */2 * * * /usr/local/apache/bin/apachectl restart >/dev/null 2>&1
```

crontab 日志

```
$ crontab -l
# m h dom mon dow    command
0 1 * * * /home/mgmt/backup && echo `date` >> /home/mgmt/backup.log
```



## 8. NFS

### Images

```
/images          172.16.1.0/24(rw, sync, fsid=0, anonuid=99, anongid=99)
/images/2008      *(rw, sync)
/images/2009      *(rw, nohide, sync)
/images/2010      *(rw, nohide, sync)
```

```
chkconfig portmap on
chkconfig nfs on
chkconfig nfslock on
chkconfig rpcidmapd on

service portmap start
service rpcidmapd start
service nfs start
service nfslock start
```

[Home](#) | [Mirror](#) | [Search](#)



## 9. 释放内存

```
sync
echo 3 > /proc/sys/vm/drop_caches
```

名称：sync  
使用权限：系统管理者  
使用方式：sync  
说明：Linux 系统中欲写入硬盘的资料有的时候会了效率起见，会写到 filesystem buffer 中，这个 buffer 是一块记忆体空间，如果欲写入硬盘的资料存于此 buffer 中，而系统又突然断电的话，那么资料就会流失了，sync 指令会将存于 buffer 中的资料强制写入硬盘中。

```
To free pagecache: echo 1 > /proc/sys/vm/drop_caches
To free dentries and inodes: echo 2 > /proc/sys/vm/drop_caches
To free pagecache, dentries and inodes: echo 3 > /proc/sys/vm/drop_caches
```



# 10. OpenSSH

## OpenSSH 安全配置

1. 禁止 root 用户登录
2. 禁止使用密码登录，使用证书登录
3.
4.
5.

history 安装,避免其他用户从history获得不安全的信息

### 10.1. history 安全

```
ln -s /dev/null .bash_history
ln -s /dev/null .mysql_history
```

### 10.2. sshd\_config

```
vim /etc/ssh/sshd_config <<end > /dev/null 2>&1
:%s/#PermitRootLogin yes/PermitRootLogin no/
:%s/#AuthorizedKeysFile/AuthorizedKeysFile/
:%s/#UseDNS yes/UseDNS no/
:wq
end
```

```
/etc/init.d/sshd reload
```



# 11. TCP/IP

## 11.1. TCP 状态

```
netstat -n | awk '/^tcp/ {++state[$NF]} END {for(key in state) print key,"\\t",state[key]}'
```

状态:	描述
CLOSED:	无连接是活动的或正在进行
LISTEN:	服务器在等待进入呼叫
SYN_RECV:	一个连接请求已经到达，等待确认
SYN_SENT:	应用已经开始，打开一个连接
ESTABLISHED:	正常数据传输状态
FIN_WAIT1:	应用说它已经完成
FIN_WAIT2:	另一边已同意释放
ITMED_WAIT:	等待所有分组死掉
CLOSING:	两边同时尝试关闭
TIME_WAIT:	另一边已初始化一个释放
LAST_ACK:	等待所有分组死掉

### 减少TIME\_WAIT状态

```
net.ipv4.tcp_syncookies = 1
```

表示开启SYN Cookies。当出现SYN等待队列溢出时，启用cookies来处理，可防范少量SYN攻击，默认为0，表示关闭；

```
net.ipv4.tcp_tw_reuse = 1
```

表示开启重用。允许将TIME-WAIT sockets重新用于新的TCP连接，默认为0，表示关闭；

```
net.ipv4.tcp_tw_recycle = 1
```

表示开启TCP连接中TIME-WAIT sockets的快速回收，默认为0，表示关闭。

```
net.ipv4.tcp_fin_timeout = 30
```

修改系统默认的 TIMEOUT 时间

```
vim /etc/sysctl.conf
```

```
net.ipv4.tcp_syncookies = 1
```

```
net.ipv4.tcp_tw_reuse = 1
```

```
net.ipv4.tcp_tw_recycle = 1
```

```
net.ipv4.tcp_fin_timeout = 30
```

## 11.2. TIME\_WAIT

```
# netstat -tna | cut -b 49- |grep TIME_WAIT | sort
```

## 11.3. SYN\_RECV / SYN\_SENT

```
# netstat -an | grep SYN | awk '{print $5}' | awk -F: '{print $1}' | sort | uniq -c | sort -nr | more
```





## 12. NTPD

```
yum install ntp -y
vim /etc/ntp.conf <<VIM > /dev/null 2>&1
:17,17s/^/server 172.16.1.10\r/
:wq
VIM
service ntpd start
chkconfig ntpd on
```



# 13. dnsmasq

```
yum install -y dnsmasq
chkconfig dnsmasq on
service dnsmasq start
```



# 14. snmp

```
yum install net-snmp -y
vim /etc/snmp/snmpd.conf <<VIM > /dev/null 2>&1
:62,62s/systemview/all/
:85,85s/^#//
:wq
VIM
service snmpd start
chkconfig snmpd on
```

[Home](#) | [Mirror](#) | [Search](#)



## 15. nrpe

```
rpm -Uvh http://download.fedora.redhat.com/pub/epel/5/x86_64/epel-release-5-4.noarch.rpm
yum install nrpe -y
yum install -y nagios-plugins-disk nagios-plugins-load nagios-plugins-ping nagios-plugins-procs
nagios-plugins-swap nagios-plugins-users

chkconfig nrpe on

vim /etc/nagios/nrpe.cfg <<VIM > /dev/null 2>&1
:79,79s/allowed_hosts=127.0.0.1/allowed_hosts=172.16.1.2/
:wq
VIM

cat >> /etc/nagios/nrpe.cfg <<EOF

#command[check_http]=/usr/lib64/nagios/plugins/check_http -I 127.0.0.1 -p 80 -u
http://www.example.com/index.html
command[check_swap]=/usr/lib64/nagios/plugins/check_swap -w 20% -c 10%
command[check_all_disks]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -e
EOF

service nrpe start
```

### NAGIOS3 Template

```
define host{
    use                generic-host
    host_name          www.example.com
    alias              www.example.com
    address            172.16.1.5
    hostgroups         mysql-servers
}
define service{
    use                generic-service ; Inherit values from a template
    host_name          www.example.com ; The name of the host the service is associated
with
    service_description PING           ; The service description
    check_command       check_ping!200.0,20%!600.0,60%   ; The command used to monitor the
service
    normal_check_interval 5           ; Check the service every 5 minutes under normal
conditions
    retry_check_interval 1            ; Re-check the service every minute until its final/hard
state is determined
}
define service{
    use                generic-service
    host_name          www.example.com
    service_description MySQL
    check_command       check_mysql_database!monitor!xmNhjDCjfYHjcSG!test
}

# NRPE disk check.
define service {
    use                generic-service
    host_name          www.example.com
    service_description nrpe-disk
    check_command       check_nrpe_larg!check_all_disks
}
define service {
```

<pre>        use         host_name         service_description         check_command     }     define service {         use         host_name         service_description         check_command     } ;define service { ;    use ;    host_name ;    service_description ;    check_command ;}; define service {     use     host_name     service_description     check_command } define service {     use     host_name     service_description     check_command } }</pre>	<pre>generic-service www.example.com nrpe-users check_nrpe_larg!check_users  generic-service www.example.com nrpe-swap check_nrpe_larg!check_swap  generic-service www.example.com nrpe-procs check_nrpe_larg!check_total_procs  generic-service www.example.com nrpe-load check_nrpe_larg!check_load  generic-service www.example.com nrpe-zombie_procs check_nrpe_larg!check_zombie_procs</pre>
---	---



## 16. vsftpd

```
yum install -y vsftpd
adduser --home-dir /www/logs/ --shell /sbin/nologin logs
echo logs >> /etc/vsftpd/chroot_list
vim /etc/vsftpd/vsftpd.conf <<VIM > /dev/null 2>&1
:%s/#chroot_list_enable=YES/chroot_list_enable=YES/
:%s/#chroot_list_file/chroot_list_file/
VIM
chkconfig vsftpd on
service vsftpd start

adduser -o --home /www --shell /sbin/nologin --uid 99 --gid 99 --group nobody www
echo "www:chen" | chpasswd
echo www > /etc/vsftpd/chroot_list
ncftp ftp://www:chen@172.16.0.1
```

[Home](#) | [Mirror](#) | [Search](#)

## 17. 安装 LAMP

```
groupadd -g 80 www
adduser -o --home /www --uid 80 --gid 80 -c "Web User" www
```

### 17.1. Apache

```
./configure --prefix=/usr/local/$HTTPD_DIR \
--with-mpm=worker \
--enable-so \
--enable-mods-shared=all \
--disable-authn_file \
--disable-authn_default \
--disable-authz_groupfile \
--disable-authz_user \
--disable-authz_default \
--disable-auth_basic \
--disable-include \
--disable-env \
--disable-autoindex \
--disable-asis \
--disable-cgi \
--disable-cgid \
--disable-negotiation \
--disable-actions \
--disable-userdir \
--disable-alias
```

```
./configure --prefix=/usr/local/httpd-2.2.16 \
--with-mpm=worker \
--enable-modules="dir mime rewrite deflate expires headers vhost_alias cache mem-cache" \
--enable-mods-shared="alias status info ssl usertrack file-cache disk-cache version " \
--disable-authn_file \
--disable-authn_default \
--disable-authz_groupfile \
--disable-authz_user \
--disable-authz_default \
--disable-auth_basic \
--disable-version \
--disable-env \
--disable-autoindex \
--disable-asis \
--disable-cgid \
--disable-cgi \
--disable-negotiation \
--disable-actions \
--disable-userdir \
--enable-so \
--enable-file-cache \
--enable-cache \
--enable-disk-cache \
--enable-mem-cache \
--enable-expires \
--enable-headers \
--enable-ssl \
--enable-info \
--enable-rewrite
```

```
make && make install
```

### 17.2. MySQL

```
./configure --prefix=/usr/local/mysql-5.1.45/ \
--enable-assembler \
--with-extra-charsets=complex \
--enable-thread-safe-client \
--with-big-tables \
--with-readline \
--with-ssl \
--with-embedded-server \
--enable-local-infile \
--with-plugins=innobase \
--with-mysqld-user=mysql \
--without-ndb-debug \
```

```
--without-debug \
--with-charset=utf8 \
--localstatedir=/data/mysql/data \
--with-collation=utf8_general_ci
make
make install
```

17.3. PHP

```
./configure --prefix=/usr/local/php-5.3.3 \
--with-config-file-path=/usr/local/php-5.3.3/etc \
--with-apxs2=/usr/local/httpd-2.2.16/bin/apxs \
--with-curl \
--with-gd \
--with-jpeg-dir=/usr/lib64 \
--with-iconv \
--with-zlib-dir \
--with-pear \
--with-xmlrpc \
--with-openssl \
--with-mysql \
--with-mysqli \
--with-pdo-mysql \
--enable-zip \
--enable-sockets \
--enable-soap \
--enable-mbstring \
--enable-magic-quotes \
--enable-inline-optimization \
--enable-xml \
--enable-ftp

export ORACLE_HOME=/usr/lib/oracle/10.2.0.3/client64
export LD_LIBRARY_PATH=/usr/lib/oracle/10.2.0.3/client64:$LD_LIBRARY_PATH
export NLS_LANG="AMERICAN_AMERICA.AL32UTF8"

./configure --prefix=/usr/local/php-5.2.14 \
--with-config-file-path=/usr/local/php-5.2.14/etc \
--with-apxs2=/usr/local/httpd-2.2.16/bin/apxs \
--with-curl \
--with-gd \
--with-jpeg-dir=/usr/lib64 \
--with-iconv \
--with-zlib-dir \
--with-pear \
--with-xmlrpc \
--with-openssl \
--with-mysql \
--with-mysqli \
--with-pdo-mysql \
--with-pdo-oci \
--enable-zip \
--enable-sockets \
--enable-soap \
--enable-mbstring \
--enable-magic-quotes \
--enable-inline-optimization \
--enable-xml \
--enable-ftp

make && make install

--with-oci8 \
--with-pdo-oci=instantclient,/usr,10.2.0.3
```

17.4. resin-pro

```
./configure \
--enable-64bit \
--enable-jni \
--enable-ssl \
--enable-debug

make && make install
```

17.5. httpd

```
AllowOverride All

<IfModule dir_module>
    DirectoryIndex index.html index.php
</IfModule>
```



```
<IfModule mpm_worker_module>
    StartServers      //Apache启动立即产生n个子进程
    MaxClients        //允许最大的客户数
    ServerLimit       //最大允许n子进程数
    MinSpareThreads   //最少有n个空线程
    MaxSpareThreads   //最多有n个空线程
    ThreadLimit        //最多有n个线程
    ThreadsPerChild   //一个子进程有n个常驻线程
    MaxRequestsPerChild //当处理的线程数达到n销毁
</IfModule>

<IfModule mpm_worker_module>
    StartServers      16
    ServerLimit       64
    ThreadLimit       256
    MaxClients        4096
    MinSpareThreads   128
    MaxSpareThreads   250
    ThreadsPerChild   256
    MaxRequestsPerChild 10000
</IfModule>
```

17.6. php

```
display_errors = Off
mysql.connect_timeout = 30
```

17.7. httpd-2.2.21 + php-5.3.8 自动安装脚本

```
#!/bin/bash
#=====
# LAMP Installing script by Neo <openunix@163.com>
# http://netkiller.sourceforge.net/
# $Id$
#=====
SRC_DIR=$(pwd)
PREFIX_DIR=/srv
EMAIL=webmaster@example.com
#=====
HTTPD_SRC='httpd-2.2.21.tar.gz'
PHP_SRC='php-5.3.8.tar.gz'
MYSQL_SRC='mysql-5.5.9.tar.gz'
MYSQL_LIBS_SRC='mysql-5.1.50.tar.gz'
MYSQL_BIN='mysql-5.1.50-linux-x86_64-glibc23.tar.gz'
JAVA_BIN='jdk-6u20-linux-x64.bin'
#=====
HTTPD_DIR=${HTTPD_SRC%.tar.gz}
PHP_DIR=${PHP_SRC%.tar.*}
MYSQL_DIR=${MYSQL_SRC%.tar.*}
MYSQL_LIBS_DIR=${MYSQL_LIBS_SRC%.tar.*}
#=====

if [ -z "$( egrep "CentOS|Redhat" /etc/issue)" ]; then
    echo 'Only for Redhat or CentOS'
    exit
fi

function clean(){
    rm -rf $HTTPD_DIR
    rm -rf $PHP_DIR
    rm -rf $MYSQL_DIR
    rm -rf $MYSQL_LIBS_DIR
}

function depend(){
    yum install gcc gcc-c++ make autoconf -y
    yum install curl-devel libmcrypt-devel gd-devel libjpeg-devel libpng-devel libXpm-devel
libxml2-devel libxslt-devel mhash-devel openldap-devel -y
    yum install ncurses-devel -y
#    yum install mysql-devel -y
    yum install libevent-devel -y
    yum install e4fsprogs -y
    yum install net-snmp-devel -y
    yum install setuptool ntsysv system-config-network-tui -y
    #rpm -Uvh http://download.fedora.redhat.com/pub/epel/5/x86_64/epel-release-5-4.noarch.rpm
}

function httpd(){
    rm -rf $HTTPD_DIR
    tar zxf $HTTPD_SRC
    cd $HTTPD_DIR

    RETVAL=$?
    [[ $RETVAL != 0 ]] && echo "Static: "$RETVAL && exit $?

    vim server/mpm/worker/worker.c <<end > /dev/null 2>&1
    :%s/#define DEFAULT_SERVER_LIMIT 16/#define DEFAULT_SERVER_LIMIT 256/
    :%s/#define DEFAULT_THREAD_LIMIT 64/#define DEFAULT_THREAD_LIMIT 1024/
    :wq
end
```

```
#define MAX_SERVER_LIMIT 20000
#define MAX_THREAD_LIMIT 20000

[[ $? -ne 0 ]] && echo "Step: server/mpm/worker/worker.c" && exit $?

./configure --prefix=$PREFIX_DIR/$HTTPD_DIR \
--with-mpm=worker \
--enable-modules="so dir mime rewrite deflate vhost_alias include" \
--enable-mods-shared="alias filter expires headers setenvif status info ssl usertrack cache mem-
cache file-cache disk-cache version mod_unique_id " \
--disable-authn_file \
--disable-authn_default \
--disable-authz_groupfile \
--disable-authz_user \
--disable-authz_default \
--disable-auth_basic \
--disable-version \
--disable-env \
--disable-autoindex \
--disable-asis \
--disable-cgid \
--disable-cgi \
--disable-negotiation \
--disable-actions \
--disable-userdir \
--enable-so \
--enable-file-cache \
--enable-cache \
--enable-disk-cache \
--enable-mem-cache \
--enable-expires \
--enable-headers \
--enable-ssl \
--enable-info \
--enable-rewrite

#--with-mpm=worker \
#--enable-so \
#--enable-mods-shared=all \
#--disable-alias \
#--disable-include \
#--disable-status \

[[ $? -ne 0 ]] && echo "Step: configure" && exit $?

make clean
make

[[ $? -ne 0 ]] && echo "Step: make" && exit $?

make install

[[ $? -ne 0 ]] && echo "Step: make install" && exit $?

cd ..
ln -s $PREFIX_DIR/$HTTPD_DIR $PREFIX_DIR/httpd
ln -s $PREFIX_DIR/httpd $PREFIX_DIR/apache

cp $PREFIX_DIR/httpd/conf/httpd.conf $PREFIX_DIR/httpd/conf/httpd.conf.original
mkdir -p /www/logs/error

vim $PREFIX_DIR/httpd/conf/httpd.conf <<end > /dev/null 2>&1
:%s/User daemon/User nobody/
:%s/Group daemon/Group nobody/
:%s/ServerAdmin you@example.com/ServerAdmin $EMAIL/
:%s!#ServerName www.example.com:80!#ServerName www.example.com:80\rServerName 127.0.0.1!
:%s#$PREFIX_DIR/$HTTPD_DIR/htdocs#/www#g
:%s/#\n AllowOverride None/#\r AllowOverride All/
:%s# DirectoryIndex index.html# DirectoryIndex index.html index.php#
:%s= #AddType text/html .shtml= AddType text/html .shtml=
:%s= #AddOutputFilter INCLUDES .shtml= AddOutputFilter INCLUDES .shtml=
:%s=AddOutputFilter INCLUDES .shtml=AddOutputFilter INCLUDES .shtml\r\r AddType application/x-
httpd-php .php .phtml\r AddType application/x-httpd-php-source .phps\r=
:%s#ErrorLog "logs/error_log"#ErrorLog "| $PREFIX_DIR/httpd/bin/rotatelogs
/www/logs/error/error.%Y-%m-%d.log 86400 480"#
:%s=#Include conf/extra/httpd-info.conf=Include conf/extra/httpd-info.conf=
:%s=#Include conf/extra/httpd-mpm.conf=Include conf/extra/httpd-mpm.conf=
:%s=#Include conf/extra/httpd-languages.conf=Include conf/extra/httpd-languages.conf=
:%s=#Include conf/extra/httpd-vhosts.conf=Include conf/extra/httpd-vhosts.conf=
:%s=#Include conf/extra/httpd-default.conf=Include conf/extra/httpd-default.conf=
:wq
end
#:%s!Listen 80!Listen 0.0.0.0:80!
#:%s!#ServerName www.example.com:80!#ServerName www.example.com:80\rServerName *!

#vim $PREFIX_DIR/httpd/conf/extra/httpd-mpm.conf <<end > /dev/null 2>&1
#:%s/<IfModule mpm_worker_module>/<IfModule mpm_worker_module>\r ServerLimit 16\r
ThreadLimit 128/
#:%s/StartServers 2/StartServers 8/
#:%s/MaxClients 150/MaxClients 2048/
#:%s/MinSpareThreads 25/MinSpareThreads 64/
#:%s/MaxSpareThreads 75/MaxSpareThreads 128/
#:%s/ThreadsPerChild 25/ThreadsPerChild 128/
#:%s/MaxRequestsPerChild 0/MaxRequestsPerChild 10000/
#wq
#end

vim $PREFIX_DIR/httpd/conf/extra/httpd-mpm.conf <<end > /dev/null 2>&1
#:%s/<IfModule mpm_worker_module>/<IfModule mpm_worker_module>\r ServerLimit 16\r
ThreadLimit 256/
#:%s/StartServers 2/StartServers 8/
#:%s/MaxClients 150/MaxClients 4096/
#:%s/MinSpareThreads 25/MinSpareThreads 64/
#:%s/MaxSpareThreads 75/MaxSpareThreads 256/
```

```
:%s/ThreadsPerChild      25/ThreadsPerChild      256/
:%s/MaxRequestsPerChild   0/MaxRequestsPerChild 10000/
:wq
end

vim $PREFIX_DIR/httpd/conf/extra/httpd-languages.conf <<end > /dev/null 2>&1
:%s/LanguagePriority en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no pl pt pt-BR ru sv
tr zh-CN zh-TW/#LanguagePriority en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no pl pt
pt-BR ru sv tr zh-CN zh-TW/
:%s/ForceLanguagePriority Prefer Fallback/#ForceLanguagePriority Prefer Fallback/
:wq
end

echo -ne "
AddDefaultCharset UTF-8
" >> $PREFIX_DIR/httpd/conf/extra/httpd-languages.conf

vim $PREFIX_DIR/httpd/conf/extra/httpd-info.conf <<end > /dev/null 2>&1
:%s/Allow from .example.com/Allow from 127.0.0.1 172.16.1 113.106.63.1/g
:wq
end

vim $PREFIX_DIR/httpd/conf/extra/httpd-default.conf <<end > /dev/null 2>&1
:%s/ServerTokens Full/ServerTokens Prod/
:wq
end

echo -ne "
$PREFIX_DIR/httpd/bin/apachectl start
" >> /etc/rc.local

}
function php(){
#ln -s /usr/lib64/mysql/* /usr/lib64/
#ln -s /srv/mysql/bin/mysql_config /usr/local/bin/

rm -rf $PHP_DIR
tar xzf $PHP_SRC
cd $PHP_DIR

./configure --prefix=$PREFIX_DIR/$PHP_DIR \
--with-config-file-path=$PREFIX_DIR/$PHP_DIR/etc \
--with-config-file-scan-dir=$PREFIX_DIR/$PHP_DIR/etc/conf.d \
--with-apxs2=$PREFIX_DIR/$HTTPD_DIR/bin/apxs \
--with-curl \
--with-gd \
--with-jpeg-dir \
--with-png-dir \
--with-freetype-dir \
--with-xpm-dir \
--with-iconv \
--with-zlib-dir \
--with-xmlrpc \
--with-openssl \
--with-mcrypt \
--with-mhash=shared \
--with-mysql=/srv/mysql-5.1.50-linux-x86_64-glibc23 \
--with-pdo-mysql=/srv/mysql-5.1.50-linux-x86_64-glibc23 \
--with-sqlite=shared \
--with-pdo-sqlite=shared \
--with-ldap=shared \
--with-snmp=shared \
--with-xsl=shared \
--with-pear \
--enable-sockets \
--enable-soap \
--enable-mbstring \
--enable-magic-quotes \
--enable-inline-optimization \
--enable-gd-native-ttf \
--enable-zip \
--enable-xml \
--enable-ftp

#--with-mysqli=/srv/mysql-5.1.50-linux-x86_64-glibc23/bin/mysql_config \
#--enable-embedded-mysqli \

#--enable-fpm \
#--with-fpm-user=nobody \
#--with-fpm-group=nobody \
#--with-mysql=/srv/mysql-5.5.9-linux2.6-x86_64 \
#--with-pdo-mysql=/srv/mysql-5.5.9-linux2.6-x86_64 \

[[ $? -ne 0 ]] && echo "Step: configure" && exit $?

#make && make test && make install
#make && make install
make
[[ $? -ne 0 ]] && echo "Step: make" && exit $?

make install
[[ $? -ne 0 ]] && echo "Step: make install" && exit $?

mkdir -p $PREFIX_DIR/$PHP_DIR/etc/conf.d
cp $PREFIX_DIR/src/$PHP_DIR/php.ini-production $PREFIX_DIR/$PHP_DIR/etc/php.ini
ln -s $PREFIX_DIR/$PHP_DIR $PREFIX_DIR/php

cp $PREFIX_DIR/$PHP_DIR/etc/php.ini $PREFIX_DIR/$PHP_DIR/etc/php.ini.original

vim $PREFIX_DIR/php/etc/php.ini <<EOF > /dev/null 2>&1
:%s!/include_path = ".:php/includes"!;include_path = ".:php/includes"\rinclude_path =
".:$PREFIX_DIR/$PHP_DIR/lib/php"!
:%s!:extension_dir = ".:":extension_dir = "$PREFIX_DIR/$PHP_DIR/lib/php/extensions":
:%s/memory_limit = 128M/memory_limit = 64M/
:%s/log errors = Off/log errors = On/
```



```
rm -rf $MEMCACHED_SRC
tar zxf $MEMCACHED_PKG
cd $MEMCACHED_SRC
./configure --prefix=$PREFIX_DIR/memcached-1.4.5
make && make install
}

function vsftpd(){
yum install -y vsftpd
adduser --home-dir /www/target/logs/ --shell /sbin/nologin --password logs.xiu.com logs
echo logs >> /etc/vsftpd/chroot_list
vim /etc/vsftpd/vsftpd.conf <<VIM > /dev/null 2>&1
:%s/#chroot_list_enable=YES/chroot_list_enable=YES/
:%s/#chroot_list_file/chroot_list_file/
VIM
chkconfig vsftpd on
service vsftpd start
}

# See how we were called.
case "$1" in
clean)
clean
;;
httpd)
httpd
;;
php)
php
;;
mysql)
if [ -f $0 ] ; then
mysql
fi
;;
depend)
depend
;;
java)
java
;;
resin)
resin
;;
profile)
profile
;;
memcache)
memcache
;;
lamp)
clean

echo #####
echo # $MYSQL_DIR Installing...
echo #####
mysql

echo #####
echo # $HTTPD_DIR Installing...
echo #####
httpd

echo #####
echo # $PHP_DIR Installing...
echo #####
php

clean
;;
vsftpd)
vsftpd
;;
optimization)
optimization
;;
*)
echo $"Usage: $0 {depend|clean}"
echo "          {httpd|php|mysql|vsftpd}"
echo "          {java|resin}"
echo "          {memcache}"
echo "          {optimization}"
echo "          {ntp|snmp|nagios|nrpe}"
RETVAL=2
;;
esac

exit $RETVAL
```

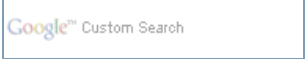


# 18. Linux 日常管理

## 日常管理的几个要点

- 1. 服务器负载 load average
- 2. 网络负载
- 3. 磁盘容量
- 4. 登录用户
- 5. 日志分析

[Home](#) | [Mirror](#) | [Search](#)



## 第 20 章 Monitor solution

### 目录

#### [1. 网络监控](#)

##### [1.1. 流量监控](#)

##### [1.2. 交换机监控](#)

#### [2. 集群监控](#)

#### [3. 服务器状态监控](#)

##### [3.1. 负载监控](#)

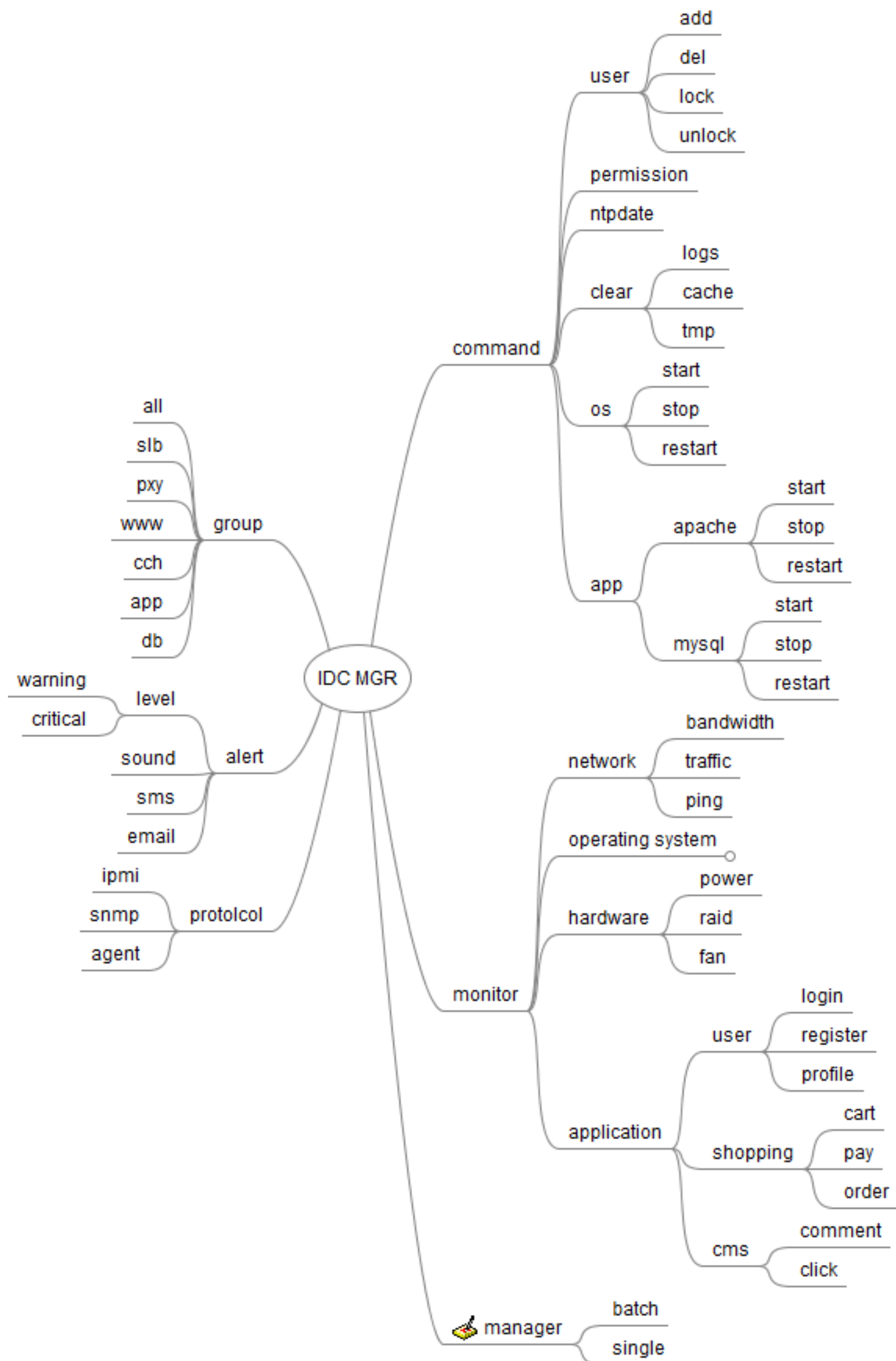
##### [3.2. 磁盘空间监控](#)

##### [3.3. 内存监控](#)

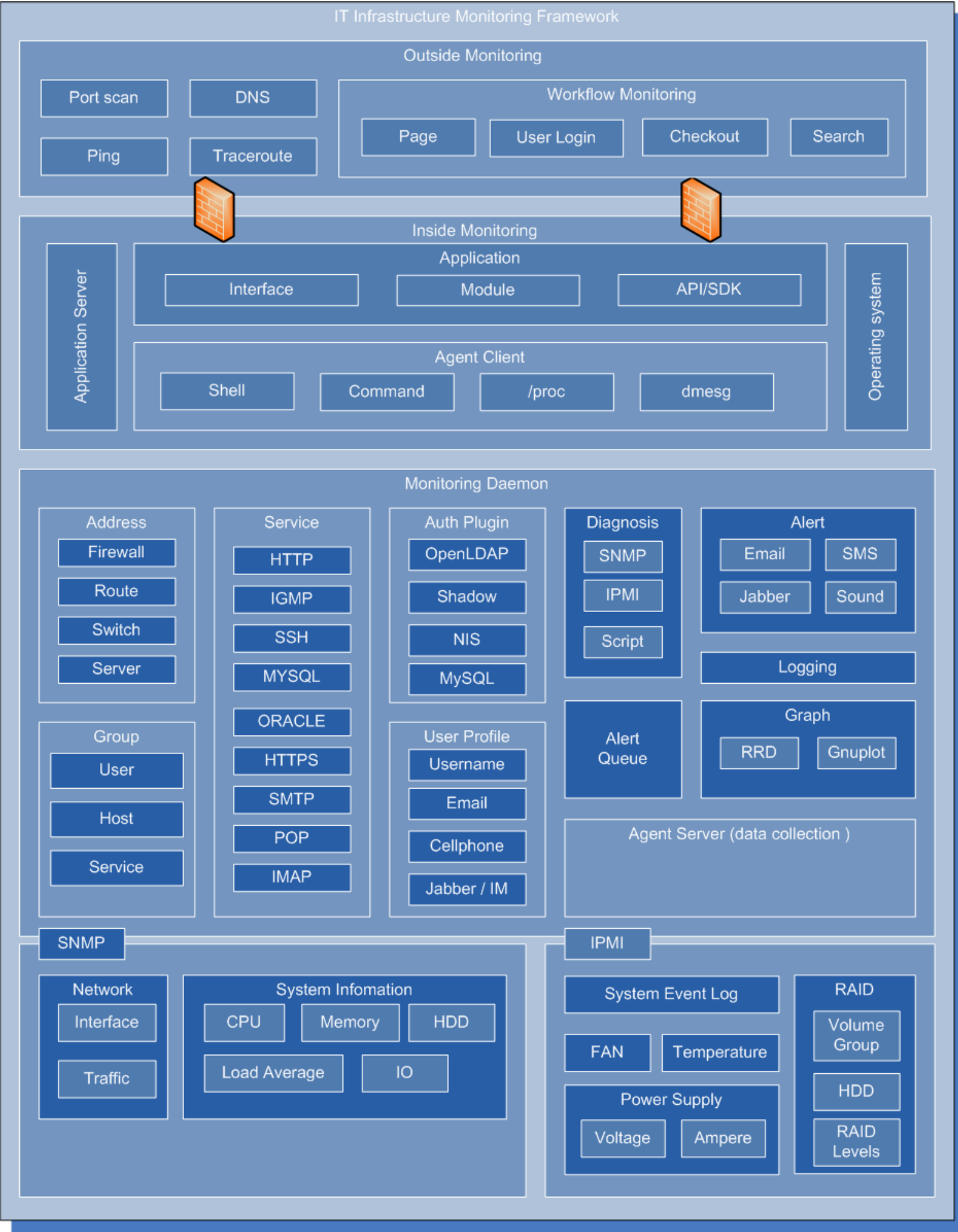
##### [3.4. CPU 监控](#)

#### [4. MySQL 监控](#)

#### [5. 网站安全与预警机制](#)







1. 网络监控

1.1. 流量监控

1.2. 交换机监控



## 2. 集群监控

在线人数

每个节点的负载情况



## 3. 服务器状态监控

3.1. 负载监控

3.2. 磁盘空间监控

3.3. 内存监控

3.4. CPU监控



## 4. MySQL 监控

MySQL 线程数

MySQL Slow日志

MySQL 节点



# 5. 网站安全与预警机制

# 第 21 章 Backup

## 目录

### [1. help](#)

#### [1.1. Task](#)

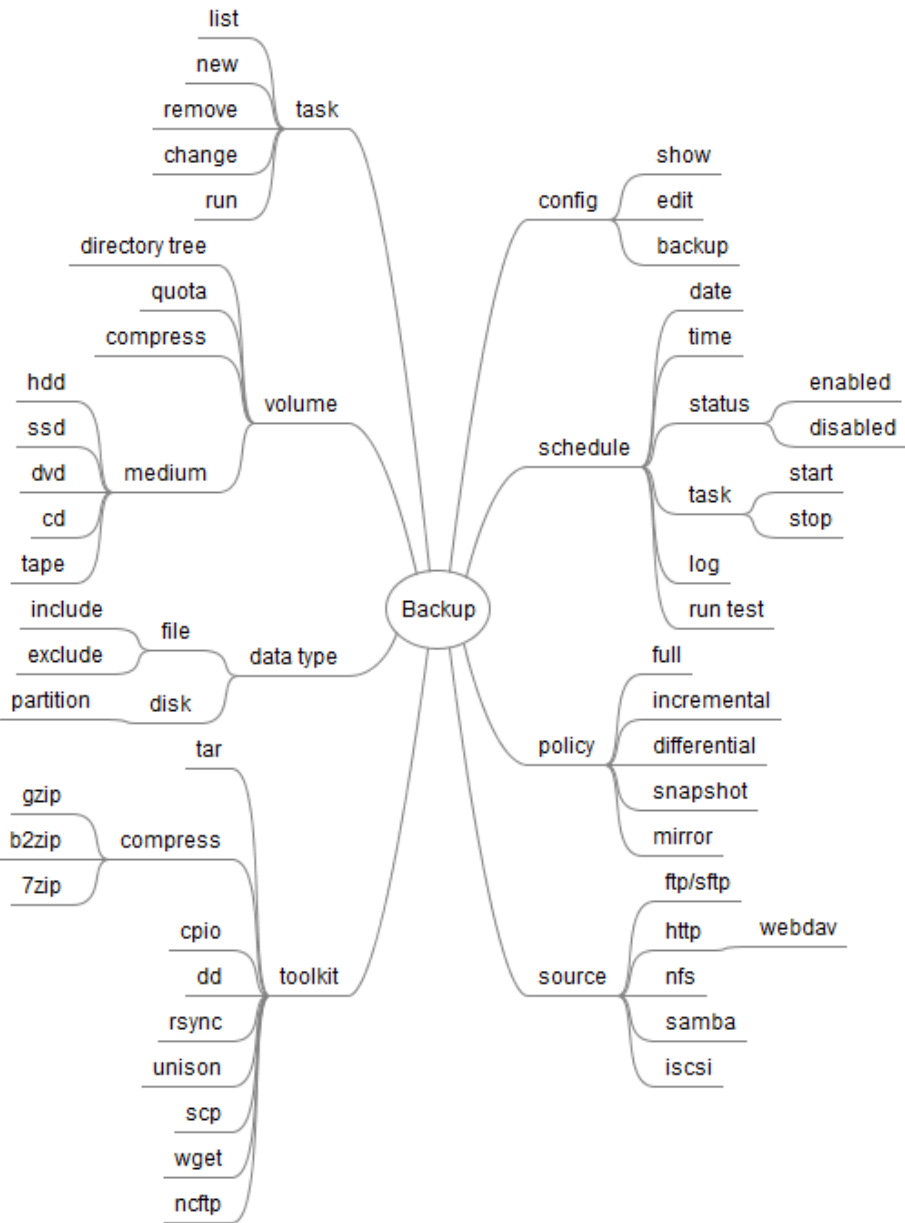
#### [1.2. Schedule](#)

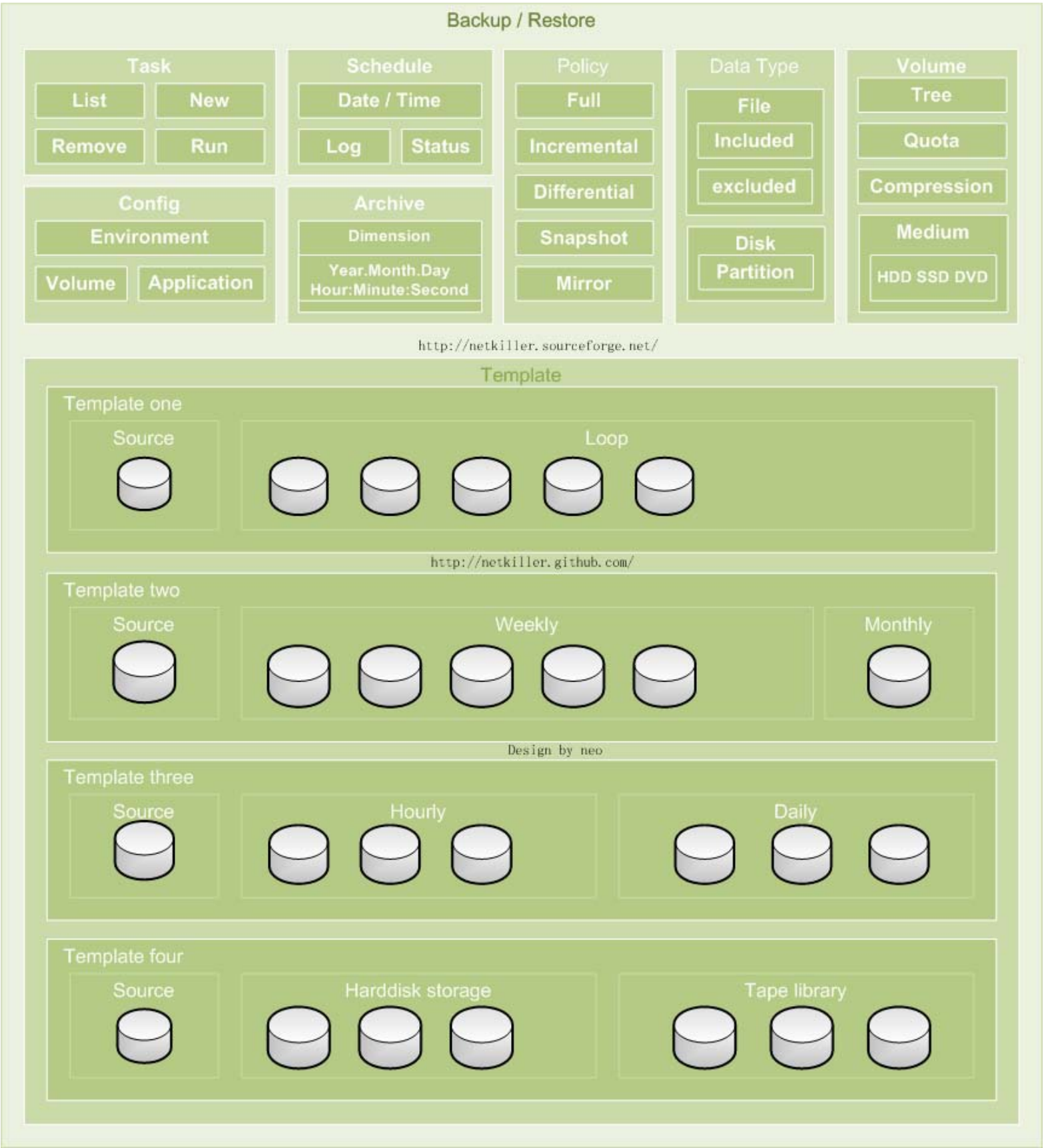
#### [1.3. Crontab](#)

### [2. 配置文件备份](#)

#### [2.1. Firewall and Switch](#)

#### [2.2. Server](#)





例 21.1. Backup program

the following is a backup program to implement my ideas in the above diagram. I have not finished yet.

```
#!/usr/bin/env python3
#/bin/env python3
#-*- coding: utf-8 -*-
#####
# Home   : http://netkiller.sf.net
# Author : Neo <openunix@163.com>
#####

try:
    import logging, configparser
    import threading
    from optparse import OptionParser, OptionGroup
    import os,io,sys
except ImportError as err:
    print("Error: %s" %(err))

class Runtime(threading.Thread):
    def __init__(self, logging):
        threading.Thread.__init__(self)
        self.logging = logging
        sys.stdin = open('stdin.log', 'r')
        sys.stdout = open('stdout.log', 'w')
        sys.stderr = open('stderr.log', 'w')
    def command(self,cmd):
        commands = {
            'rsync': 'rsync -auzvP',
```

```

        'sftp' : 'sftp',
        'scp'  : 'scp -r',
        'cp'   : 'cp -r'
    }
    return commands[cmd]
def policy(self, policy):
    policies = {
        'full': 'rsync -az',
        'mirror': 'rsync -auz --delete',
        'differential': 'rsync -auz --delete',
        'incremental': 'rsync -auz',
        'clone': 'dd',
        'copy.cp': 'cp -au',
        'copy.cp.backup': "cp --suffix=$(date '+.%Y-%m-%d.%H:%M:%S') "
        'copy.scp': 'scp -a',
        'mirror.ftp': 'wget -m',
        'mirror.http': 'wget -m',
        'archive.zip': 'zip',
        'archive.7zip': '7zip',
        'archive.cpio': 'cpio',
        'archive.tar': 'tar zcvf',
        'snapshot': ''
    }
    return policies[policy]
def execute(self, cfg):
    print( self.policy('mirror'))
    #command = self.command(cfg['cmd'])
    command = self.policy(cfg['policy']) + ' ' + cfg['source'] + ' ' + cfg['target'] + ' >> stdout.log'
    self.logging.debug(command)
    os.system(command)
class Task():
    def __init__(self, logging):
        self.logging = logging
        self.config = configparser.SafeConfigParser()
        cfg='task.cfg'
        self.config.read(cfg)
    def list(self):
        for section in self.config.sections():
            print(section)
    def new(self):
        pass
    def remove(self):
        pass
    def change(self):
        pass
    def run(self, section):
        try:
            #print(cfg)
            cfg = self.config.items(section)
            r = Runtime(self.logging)
            r.execute(dict(cfg))
        except configparser.NoSectionError as err:
            print(err)
    def show(self, section):
        for item in self.config.items(section):
            #k,v = item
            print("%s: %s" %(item))
    def get(self, section):
        return self.config.items(section)
class Schedule():
    def __init__(self, logging):
        self.logging = logging
        self.config = configparser.SafeConfigParser()
        cfg='schedule.cfg'
        self.config.read(cfg)
    def list(self):
        for section in self.config.sections():
            print(section)
    def show(self, section):
        for item in self.config.items(section):
            print("%s: %s" %(item))
    def new(self):
        pass
    def remove(self):
        pass
    def change(self):
        system('backup.cron')
    def status(self):
        pass
    def run(self, section):
        threads = []
        t = Task(self.logging)
        #t.run(task)
        for task, status in self.config.items(section):
            if status :
                cfg = t.get(task)
                r = Runtime(dict(cfg))
                r.setName('Thread-' + task)
                threads.append(r)
        for t in threads:
            #print(t.getName())
            self.logging.info(t.getName())
            t.start()
            t.join()
class Volume():
    pass
class Backup():
    def __init__(self):
        self.config = {}
        #self.config['logfile'] = 'backup.log'

        usage = "usage: %prog [options] arg1 arg2"
        self.parser = OptionParser(usage)
        self.parser.add_option("-f", "--file", dest="filename",
            help="write report to FILE", metavar="FILE")
        self.parser.add_option("-q", "--quiet",
            action="store_false", dest="verbose", default=True,
            help="don't print status messages to stdout")
        self.parser.add_option('', '--config', dest="config", help='Read configuration options from file.',
            default='backup.cfg')

        group = OptionGroup(self.parser, "arg1",
            "arg1: task | schedule")
        self.parser.add_option_group(group)
        group = OptionGroup(self.parser, 'arg2', 'arg2: list | new | remove | show')

```



```

self.parser.add_option_group(group)

self.parser.add_option('-v','--version',action='store_true', help='print version number')
self.parser.add_option('-d','--daemon', dest='daemon', action='store_true', help='run as daemon')
self.parser.add_option('', '--logfile', help='logs file.', default='backup.log')

(options, args) = self.parser.parse_args()
self.configure(options)

try:
    logging.basicConfig(level=logging.NOTSET,
                        format='%(asctime)s %(levelname)-8s %(message)s',
                        datefmt='%Y-%m-%d %H:%M:%S',
                        filename=self.config['environment']['logfile'],
                        filemode='a')
    self.logging = logging.getLogger()
except AttributeError as err:
    print("Error: %s %s" %(err, self.config['environment']['logfile']))
    sys.exit(2)
pass

def configure(self,options):
    if options.config:
        cpr = configparser.SafeConfigParser()
        cpr.read(options.config)
        for sect in cpr.sections():
            self.config[sect] = dict(cpr.items(sect))
            #for (key,value) in cpr.items(sect):
            #    self.config[key] = value
        #print(self.config)

def task(self, args):
    try:
        t = Task(self.logging)
        if len(args) <= 1:
            t.list()
        elif args[1] == 'list':
            t.list()
        elif args[1] == 'run':
            if len(args) == 3:
                t.run(args[2])
            else:
                t.list()
        elif args[1] == 'show':
            if len(args) == 3:
                t.show(args[2])
            else:
                t.list()
        else:
            t.list()
    except IOError as err:
        print(err)
    except configparser.NoSectionError as err:
        t.list()
        print(err)
        self.logging.error(err)

def schedule(self,args):
    try:
        s = Schedule(self.logging)
        if len(args) <= 1:
            s.list()
        elif args[1] == 'list':
            s.list()
        elif args[1] == 'show':
            if len(args) == 3:
                s.show(args[2])
            else:
                s.list()
        elif args[1] == 'run':
            if len(args) == 3:
                s.run(args[2])
            else:
                s.list()
        else:
            s.list()
    except configparser.NoSectionError as err:
        s.list()
        self.logging.error(err)
        print(err)

def usage(self):
    self.parser.print_help()

def main(self):
    (options, args) = self.parser.parse_args()
    if options.daemon:
        pid = os.fork()
        if pid > 0:
            self.logging.info('daemon is ok')
            sys.exit(0)
    if not args:
        self.usage()
    elif args[0] == 'task':
        self.task(args)
        self.logging.debug('Task')
    elif args[0] == 'schedule':
        self.schedule(args)
        self.logging.debug('Schedule')
    else:
        print('')

if __name__ == '__main__':
    try:
        backup = Backup()
        backup.main()
    except KeyboardInterrupt:
        print ("Ctrl+C Pressed. Shutting down.")

```

# 1. help

## 1.1. Task

list task

```
$ backup task list
www
database
test
test1
test2
```

show task

```
$ backup task show www
policy: mirror
source: /www/www.example.dev/*
target: /tmp/www
exclude: .svn
include: *
```

run task

```
$ backup task run www
```

## 1.2. Schedule

the backup program has four schedule task, actually you can add more.

```
$ ./backup schedule list
hourly
daily
weekly
monthly

$ ./backup schedule show hourly
www: True
mrtg: True

$ ./backup schedule run hourly
```

## 1.3. Crontab

crontab -l

```
17 *      * * *   test -x /srv/sbin/backup || ( backup schedule run hourly )
25 6      * * *   test -x /srv/sbin/backup || ( backup schedule run daily   )
47 6      * * 7   test -x /srv/sbin/backup || ( backup schedule run weekly  )
52 6      1 * *   test -x /srv/sbin/backup || ( backup schedule run monthly )

# m h  dom mon dow   command
*/30 * * * * /srv/sbin/backup task run www
```



## 2. 配置文件备份

### 2.1. Firewall and Switch

Tftp

### 2.2. Server

[Home](#) | [Mirror](#) | [Search](#)



# 第 22 章 常用工具

## 目录

- [1. IP Zone](#)
- [2. What is my IP address?](#)
  - [2.1.](#)
- [3. IP Tracking](#)
- [4. 自动化脚本](#)
  - [4.1. Install Script](#)
  - [4.2. FTP User Script](#)
  - [4.3. SVN User Script](#)

## 1. IP Zone

<http://www.ipdeny.com>



## 2. What is my IP address?

<http://www.ipaddressworld.com/>

curl http://members.3322.org/dyndns/getip



### 3. IP Tracking

[http://www.ip-adress.com/ip\\_tracer/](http://www.ip-adress.com/ip_tracer/)

[Home](#) | [Mirror](#) | [Search](#)

## 4. 自动化脚本

### 4.1. Install Script

```
#!/bin/bash
#=====
# LAMP Installing script by Neo <openunix@163.com>
# http://netkiller.sourceforge.net/
# $Id$
#=====
SRC_DIR=$(pwd)
PREFIX_DIR=/srv
HTTPD_SRC=httpd-2.2.17.tar.gz
PHP_SRC=php-5.3.5.tar.gz
MYSQL_SRC='mysql-5.5.9.tar.gz'
MYSQL_LIBS_SRC='mysql-5.1.50.tar.gz'
MYSQL_BIN='mysql-5.1.50-linux-x86_64-glibc23.tar.gz'
JAVA_BIN='jdk-6u20-linux-x64.bin'
#=====
HTTPD_DIR=${HTTPD_SRC%.tar.gz}
PHP_DIR=${PHP_SRC%.tar.*}
MYSQL_DIR=${MYSQL_SRC%.tar.*}
MYSQL_LIBS_DIR=${MYSQL_LIBS_SRC%.tar.*}

#=====

if [ -z "$( egrep "CentOS|Redhat" /etc/issue)" ]; then
    echo 'Only for Redhat or CentOS'
    exit
fi

function clean(){
    rm -rf $HTTPD_DIR
    rm -rf $PHP_DIR
    rm -rf $MYSQL_DIR
    rm -rf $MYSQL_LIBS_DIR
}

function tools(){
    yum install wget telnet tcpdump -y
}

function ntp(){
    yum install ntp -y
}

vim /etc/ntp.conf <<VIM > /dev/null 2>&1
:17,17s/^/server 172.16.1.10\r/
:wq
VIM

    service ntpd start
    chkconfig ntpd on
}

function snmp (){
    yum install net-snmp -y

vim /etc/snmp/snmpd.conf <<VIM > /dev/null 2>&1
:62,62s/systemview/all/
:85,85s/^#//
:wq
VIM

    service snmpd start
    chkconfig snmpd on
}

function depend(){
    yum install gcc gcc-c++ make autoconf -y
    yum install curl-devel libmcrypt-devel gd-devel libjpeg-devel libpng-devel libXpm-devel
libxml2-devel libxslt-devel openldap-devel -y
    yum install ncurses-devel -y
#    yum install mysql-devel -y
    yum install libevent-devel -y

    yum install e4fsprogs -y
    yum install net-snmp-devel -y
    yum install setuptool ntsysv system-config-network-tui -y
    rpm -Uvh http://download.fedora.redhat.com/pub/epel/5/x86_64/epel-release-5-4.noarch.rpm
}

function mysql(){
    rm -rf $MYSQL_DIR
    tar xzf $MYSQL_SRC
    cd $MYSQL_DIR
```

```

./configure \
--prefix=$PREFIX_DIR/$MYSQL_DIR \
--with-mysqld-user=mysql \
--with-unix-socket-path=/tmp/mysql.sock \
--with-charset=utf8 \
--with-collation=utf8_general_ci \
--with-extra-charsets=complex \
--with-big-tables \
--with-readline \
--with-ssl \
--with-embedded-server \
--with-plugins=innobase \
--with-mysqld-user=mysql \
--without-ndb-debug \
--without-debug \
--without-bench \
--without-docs \
--localstatedir=/data/mysql/data \
--enable-asm \
--enable-thread-safe-client \
--enable-local-infile \

make clean
make && make install
cd ..

ln -s $PREFIX_DIR/$MYSQL_DIR $PREFIX_DIR/mysql
$PREFIX_DIR/$MYSQL_DIR/bin/mysql_install_db --user=mysql
chown mysql:mysql -R $PREFIX_DIR/$MYSQL_DIR
cp $PREFIX_DIR/src/$MYSQL_DIR/support-files/mysql.server /etc/init.d/mysql
/etc/init.d/mysql start
$PREFIX_DIR/$MYSQL_DIR/bin/mysqladmin -u root password '9S5wxCVPMY'
}
function httpd(){
rm -rf $HTTPD_DIR
tar xzf $HTTPD_SRC
cd $HTTPD_DIR
./configure --prefix=$PREFIX_DIR/$HTTPD_DIR \
--with-mpm=worker \
--enable-modules="so dir mime rewrite deflate vhost_alias include" \
--enable-mods-shared="alias filter expires headers setenvif status info ssl usertrack cache mem-
cache file-cache disk-cache version mod_unique_id " \
--disable-authn_file \
--disable-authn_default \
--disable-authz_groupfile \
--disable-authz_user \
--disable-authz_default \
--disable-auth_basic \
--disable-version \
--disable-env \
--disable-autoindex \
--disable-asis \
--disable-cgid \
--disable-cgi \
--disable-negotiation \
--disable-actions \
--disable-userdir \
--enable-so \
--enable-file-cache \
--enable-cache \
--enable-disk-cache \
--enable-mem-cache \
--enable-expire \
--enable-headers \
--enable-ssl \
--enable-info \
--enable-rewrite

#--with-mpm=worker \
#--enable-so \
#--enable-mods-shared=all \
#--disable-alias \
#--disable-include \
#--disable-status \

make clean
make && make install
cd ..
ln -s $PREFIX_DIR/$HTTPD_DIR $PREFIX_DIR/httpd
ln -s $PREFIX_DIR/httpd $PREFIX_DIR/apache

cp $PREFIX_DIR/httpd/conf/httpd.conf $PREFIX_DIR/httpd/conf/httpd.conf.old

vim $PREFIX_DIR/httpd/conf/httpd.conf <<end > /dev/null 2>&1
:s/User daemon/User nobody/
:s/Group daemon/Group nobody/
:s!#ServerName www.example.com:80!#ServerName www.example.com:80\rServerName *!
:s# $PREFIX_DIR/httpd-2.2.16/htdocs#/www#g
:s/#\n AllowOverride None/#\r AllowOverride All/
:s# DirectoryIndex index.html# DirectoryIndex index.html index.php#
:s= #AddType text/html .shtml= AddType text/html .shtml=
:s= #AddOutputFilter INCLUDES .shtml= AddOutputFilter INCLUDES .shtml=
:s=AddOutputFilter INCLUDES .shtml=AddOutputFilter INCLUDES .shtml\r\r AddType application/x-
httpd-php .php .phtml\r AddType application/x-httpd-php-source .phps\r=
:s=#Include conf/extra/httpd-mpm.conf=Include conf/extra/httpd-mpm.conf=
:s=#Include conf/extra/httpd-languages.conf=Include conf/extra/httpd-languages.conf=
:s=#Include conf/extra/httpd-vhosts.conf=Include conf/extra/httpd-vhosts.conf=
:wq
end

vim $PREFIX_DIR/httpd/conf/extra/httpd-mpm.conf <<end > /dev/null 2>&1
:s/<IfModule mpm_worker_module>/<IfModule mpm_worker_module>\r ServerLimit 16\r
ThreadLimit 128/
:s/StartServers 2/StartServers 8/
:s/MaxClients 150/MaxClients 2048/

```



```
:%s/MinSpareThreads      25/MinSpareThreads      64/
:%s/MaxSpareThreads      75/MaxSpareThreads      128/
:%s/ThreadsPerChild      25/ThreadsPerChild      128/
:%s/MaxRequestsPerChild  0/MaxRequestsPerChild 10000/
:wq
end

#vim $PREFIX_DIR/httpd/conf/extra/httpd-mpm.conf <<end > /dev/null 2>&1
#:%s/<IfModule mpm_worker_module>/<IfModule mpm_worker_module>\r      ServerLimit      16\r
ThreadLimit      256/
#:%s/StartServers      2/StartServers      8/
#:%s/MaxClients      150/MaxClients      4096/
#:%s/MinSpareThreads      25/MinSpareThreads      64/
#:%s/MaxSpareThreads      75/MaxSpareThreads      256/
#:%s/ThreadsPerChild      25/ThreadsPerChild      256/
#:%s/MaxRequestsPerChild  0/MaxRequestsPerChild 10000/
#wq
#end
```

```
vim $PREFIX_DIR/httpd/conf/extra/httpd-languages.conf <<end > /dev/null 2>&1
:%s/LanguagePriority en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no pl pt pt-BR ru sv
tr zh-CN zh-TW/#LanguagePriority en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no pl pt
pt-BR ru sv tr zh-CN zh-TW/
:%s/ForceLanguagePriority Prefer Fallback/#ForceLanguagePriority Prefer Fallback/
:wq
end
```

```
echo -ne "
AddDefaultCharset UTF-8
" >> $PREFIX_DIR/httpd/conf/extra/httpd-languages.conf
```

```
echo -ne "
$PREFIX_DIR/httpd/bin/apachectl start
" >> /etc/rc.local
```

```
}
function php(){
#configure: error: Cannot find MySQL header files under
#ln -s $PREFIX_DIR/mysql-5.1.45/bin/mysql_config $PREFIX_DIR/bin/
#rm -rf $MYSQL_LIBS_DIR
#tar zxf $MYSQL_LIBS_SRC
# or
# rpm -ivh MySQL-devel-5.1.50-1.glibc23.x86_64.rpm
# rpm -ivh MySQL-shared-5.1.50-1.glibc23.x86_64.rpm

#ln -s /usr/lib64/mysql/* /usr/lib64/
rm -rf $PHP_DIR
tar zxf $PHP_SRC
cd $PHP_DIR
```

```
./configure --prefix=$PREFIX_DIR/$PHP_DIR \
--with-config-file-path=$PREFIX_DIR/$PHP_DIR/etc \
--with-config-file-scan-dir=$PREFIX_DIR/$PHP_DIR/etc/conf.d \
--with-apxs2=$PREFIX_DIR/$HTTPD_DIR/bin/apxs \
--with-curl \
--with-gd \
--with-jpeg-dir=/usr/lib64 \
--with-png-dir \
--with-freetype-dir \
--with-xpm-dir \
--with-iconv \
--with-zlib-dir \
--with-xmlrpc \
--with-ldap \
--with-openssl \
--with-mcrypt \
--with-mhash \
--with-mysql=/srv/mysql-5.5.9-linux2.6-x86_64 \
--with-mysqli \
--with-pdo-mysql=/srv/mysql-5.5.9-linux2.6-x86_64 \
--with-snmp \
--with-xsl \
--with-pear \
--enable-sockets \
--enable-soap \
--enable-mbstring \
--enable-magic-quotes \
--enable-inline-optimization \
--enable-gd-native-ttf \
--enable-zip \
--enable-xml \
--enable-ftp
```

```
#--enable-fpm \
#--with-fpm-user=nobody \
#--with-fpm-group=nobody \
```

```
#make && make test && make install
make && make install
cp $PREFIX_DIR/src/$PHP_DIR/php.ini-production $PREFIX_DIR/$PHP_DIR/etc/php.ini
ln -s $PREFIX_DIR/$PHP_DIR $PREFIX_DIR/php
```

```
cp $PREFIX_DIR/$PHP_DIR/etc/php.ini $PREFIX_DIR/$PHP_DIR/etc/php.ini.old
```

```
vim $PREFIX_DIR/php/etc/php.ini <<EOF > /dev/null 2>&1
:%s!;include_path = ".:/php/includes"!;include_path = ".:/php/includes"\rinclude_path =
".:$PREFIX_DIR/$PHP_DIR/lib/php"!
:%s:extension_dir = "./:":extension_dir = "$PREFIX_DIR/$PHP_DIR/lib/php/extensions":
:%s/output_buffering = Off/output_buffering = On/
:%s/memory_limit = 16M/memory_limit = 1/
:%s/display_errors = On/display_errors = Off/
:%s/log_errors = Off/log_errors = On/
:%s#error_log = php errors.log#error_log = /www/logs/php errors.log#
```

```
:%s/register_long_arrays = On/register_long_arrays = Off/
:%s#/open_basedir = #open_basedir = /www/:/tmp/#
:%s/upload_max_filesize = 2M/upload_max_filesize = 8M/
:%s/mysql.allow_persistent = On/mysql.allow_persistent = Off/
:wq
EOF
#:%s/allow_call_time_pass_reference = On/allow_call_time_pass_reference = Off/

    read -p "php memcache module? [y/n]" -n 1 key
    echo
    if [ $key = 'y' ]; then
$PREFIX_DIR/php/bin/pecl install memcache
cp $PREFIX_DIR/$PHP_DIR/lib/php/extensions/no-debug-zts-*/memcache.so
$PREFIX_DIR/$PHP_DIR/lib/php/extensions/
vim $PREFIX_DIR/php/etc/php.ini <<EOF > /dev/null 2>&1
:%s#;extension=php_zip.dll#;extension=php_zip.dll\rextension=memcache.so#
:wq
EOF
        echo 'memcache.default_timeout_ms=30' >> $PREFIX_DIR/php/etc/php.ini
    fi

    read -p "php APC module? [y/n]" -n 1 key
    echo
    if [ $key = 'y' ]; then
$PREFIX_DIR/php/bin/pecl install apc
cp $PREFIX_DIR/$PHP_DIR/lib/php/extensions/no-debug-zts-*/apc.so
$PREFIX_DIR/$PHP_DIR/lib/php/extensions/
vim $PREFIX_DIR/php/etc/php.ini <<EOF > /dev/null 2>&1
:%s#;extension=php_zip.dll#;extension=php_zip.dll\rextension=apc.so#
:wq
EOF
        fi

}

function java(){
    JAVA_DIR=${JAVA_BIN%.bin}
    #yum install java-1.6.0-openjdk -y
    chmod +x $JAVA_BIN
    ./ $JAVA_BIN
    mv $JAVA_DIR ..
    ln -s $PREFIX_DIR/$JAVA_DIR $PREFIX_DIR/java
echo -ne '
export JAVA_HOME=$PREFIX_DIR/java
export CLASSPATH=$JAVA_HOME/lib:$JAVA_HOME/jre/lib:$CLASSPATH
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$HOMR/bin:$PATH
export
PATH=$PATH:$PREFIX_DIR/httpd/bin:$PREFIX_DIR/php/bin:$PREFIX_DIR/mysql/bin:$PREFIX_DIR/resin/bin:$PR
' >> /etc/profile
}

function resin(){
    tar xzf resin-4.0.6.tar.gz
    mv resin-4.0.6 $PREFIX_DIR/
    echo 'export RESIN_HOME=$PREFIX_DIR/resin' >> /etc/profile
}

function memcache(){
    MEMCACHED_PKG=memcached-1.4.5.tar.gz
    MEMCACHED_SRC=memcached-1.4.5
    rm -rf $MEMCACHED_SRC
    tar xzf $MEMCACHED_PKG
    cd $MEMCACHED_SRC
    ./configure --prefix=$PREFIX_DIR/memcached-1.4.5
    make && make install
}

function optimization(){
    #echo >> /etc/profile
    #echo "export JAVA_HOME=$PREFIX_DIR/java" >> /etc/profile

#vim /etc/profile <<EOF > /dev/null 2>&1
#:#31,3ls:^:ulimit -SHn 20480 > /dev/null 2>&1\r:
#:#wq
#EOF

cat >> /etc/security/limits.conf <<EOF
root    soft  nofile 20480
root    hard  nofile 65536
nobody  soft  nofile 20480
nobody  hard  nofile 65536
EOF

cat >> /etc/sysctl.conf <<EOF
net.ipv4.ip_local_port_range = 1024 65500
net.ipv4.tcp_max_syn_backlog = 8192
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_keepalive_intvl = 30
net.ipv4.tcp_keepalive_probes = 3
net.ipv4.tcp_keepalive_time = 1200
net.ipv4.tcp_max_tw_buckets = 4096
EOF
/sbin/sysctl -p

1.net.ipv4.tcp_syncookies = 1
2.net.ipv4.tcp_tw_reuse = 1
3.net.ipv4.tcp_tw_recycle = 1
4.net.ipv4.tcp fin timeout = 30
```

```

#:54,88s/^/#/
#:57,57s/^#//
#:65,65s/^#//
#:68,68s/^#//
#:74,75s/^#//
#:80,80s/^#//
#:83,84s/^#//
#:85,87s/^#//
#:58,58s/^#//

}

function nagios(){
    yum install nagios
}

function nrpe(){
    yum install nrpe -y
    yum install -y nagios-plugins-disk nagios-plugins-load nagios-plugins-ping nagios-
plugins-procs nagios-plugins-swap nagios-plugins-users

    chkconfig nrpe on

    cat >> /etc/nagios/nrpe.cfg <<EOF

#command[check_http]=usr/lib64/nagios/plugins/check_http -I 127.0.0.1 -p 80 -u
http://www.example.com/index.html
command[check_swap]=usr/lib64/nagios/plugins/check_swap -w 20% -c 10%
command[check_all_disks]=usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -e
EOF

    service nrpe start
}

security(){

vim /etc/ssh/sshd_config <<VIM > /dev/null 2>&1
:%s/#PermitRootLogin yes/PermitRootLogin no/
:%s/#AuthorizedKeysFile/AuthorizedKeysFile/
:wq
VIM
/etc/init.d/sshd restart

vim /etc/sudoers <<VIM > /dev/null 2>&1
:86,86s/^# //
:wq!
VIM

vim /etc/group <<VIM > /dev/null 2>&1
:11,11s/$/,neo/
:wq
VIM

}

function sshkey(){

mkdir .ssh
chmod 755 .ssh

cat >> .ssh/authorized_keys <<EOF
ssh-rsa
AAAAB3NzaClyc2EAAAABI/BtYR65XoC0B4luyK/ZvPqkS4y8JNYnRGZ8aav5aaYj6xO+/dRATaL2S/soo+i05AFCYtD7xUMKhHrh.
neo.chen@example.com
EOF
chmod 644 .ssh/authorized_keys

}

function vsftpd(){
    yum install -y vsftpd
    adduser --home-dir /www/target/logs/ --shell /sbin/nologin --password your_passwd logs
    echo logs >> /etc/vsftpd/chroot_list
    vim /etc/vsftpd/vsftpd.conf <<VIM > /dev/null 2>&1
:%s/#chroot_list_enable=YES/chroot_list_enable=YES/
:%s/#chroot_list_file/chroot_list_file/
VIM
    chkconfig vsftpd on
    service vsftpd start
}

# See how we were called.
case "$1" in
    clean)
        clean
        ;;
    httpd)
        httpd
        ;;
    php)
        php
        ;;
    mysql)
        if [ -f $0 ] ; then
            mysql
        fi
        ;;
    depend)
        depend
        snmp
        nrpe
        ntp

```

```

        optimization
        ;;
java)
    java
    ;;
resin)
    resin
    ;;
profile)
    profile
    ;;
memcache)
    memcache
    ;;
lamp)
    clean

    echo #####
    echo # $MYSQL_DIR Installing...
    echo #####
    mysql

    echo #####
    echo # $HTTPD_DIR Installing...
    echo #####
    httpd

    echo #####
    echo # $PHP_DIR Installing...
    echo #####
    php

    clean
    ;;
vsftpd)
    vsftpd
    ;;
optimization)
    optimization
    ;;
ntpd)
    ntpd
    ;;
snmp)
    snmp
    ;;
nrpe)
    nrpe
    ;;
*)
    echo $"Usage: $0 {depend|clean}"
    echo "                {httpd|php|mysql|vsftpd}"
    echo "                {java|resin}"
    echo "                {memcache}"
    echo "                {optimization}"
    echo "                {ntp|snmp|nrpe}"
    RETVAL=2
    ;;
esac

exit $RETVAL
```

4.2. FTP User Script

```
#!/bin/bash
#####
# Netkiller series tools
# Home: http://netkiller.sf.net
# Author: Neo Chen<openunix@163.com>
#####

HOME_DIR=/www
shell=/sbin/nologin
uid=$(id -u nobody)
gid=$(id -g nobody)
group='nobody'

addftpuser(){
    if [ -z $2 ]; then
        usage
    else
        local HOME_DIR=$2
    fi
    if [ -z $3 ]; then
        usage
    else
        local user=$3
    fi

    if [ -z $4 ]; then
        local passwd=$(cat /dev/urandom | tr -dc [:alnum:] | head -c 20)
    else
        local passwd=$4
    fi
}
```

```

    adduser -o --home "$HOME_DIR" --shell "$shell" --uid "$uid" --gid "$gid" --group "$group"
$user
    echo "$user:$passwd" | chpasswd

    echo $user >> /etc/vsftpd/chroot_list
    echo "ftp://$user:$passwd@$(hostname)"
}
delftpuser(){
    local user=$1
    if [ -z $user ]; then
        usage
    else
        userdel $user
    fi
}
listftpuser(){
    cat /etc/passwd | grep `id -u nobody`
}
chroot_list(){
    cat /etc/vsftpd/chroot_list
}
usage(){
    echo $"Usage: $0 add <directory> <username>"
    echo $"          $0 {list|del} <username>"
    echo $"          $0 chroot_list"
    echo "Example:"
    echo "          $0 add /home/test test"
    echo "          $0 add /home/test test testpassword"
    echo "          $0 list test"
    echo "          $0 del test"

    echo "E-mail bug reports to: <openunix@163.com>."
    echo "Be sure to include the word ``ftpuser`` somewhere in the ``Subject:`` field."

    exit 2
}
case "$1" in
    list)
        listftpuser
        ;;
    add)
        addftpuser $2
        ;;
    del)
        delftpuser $2
        ;;
    chroot_list)
        chroot_list
        ;;
    *)
        usage
        exit 1
esac

```

### 4.3. SVN User Script

```

#!/bin/bash
#####
# Netkiller series tools
# Home: http://netkiller.sf.net
# Author: Neo Chen<openunix@163.com>
#####
SVNROOT=/srv/svnroot/zoshow
adduser(){
    if [ -z $1 ]; then
        usage
    else
        local user=$1
    fi
    if [ -z $2 ]; then
        passwd=$(cat /dev/urandom | tr -dc [:alnum:] | head -c 10)
    else
        local passwd=$2
    fi

    echo $user = $passwd
    echo "$user = $passwd" >> $SVNROOT/conf/passwd
}
deluser(){
    local user=$1
    if [ -z $user ]; then
        usage
    else
        ed -s $SVNROOT/conf/passwd <<EOF
/^$user = /
d
wq
EOF
    fi
}
list(){
    cat $SVNROOT/conf/passwd
}
usage(){
    echo $"Usage: $0 {list|add|del} username"
}

```

```
        exit 2
    }
    case "$1" in
        list)
            list
            ;;
        add)
            adduser $2 $3
            ;;
        del)
            deluser $2
            ;;
        restart)
            stop
            start
            ;;
        condrestart)
            condrestart
            ;;
        *)
            usage
            exit 1
    esac
```



# 第 23 章 服务器安全

## 目录

- [1. 防火墙](#)
- [2. Web Server](#)
  - [2.1. 目录访问控制](#)
  - [2.2. fastcgi 安全](#)
  - [2.3. php 函数安全](#)
  - [2.4. 服务器版本信息](#)

## 1. 防火墙

```
netstat -an|grep tcp|grep 8080|grep -v "xx.xx.xx.xx"|grep ESTABLISHED|awk '{print $5}'|sed 's/\.:\:ffff\:\/\/g'|awk -F ":" '{print $1}'|sort -r|uniq -c|sort -nk 1|awk '{if ($1>20)print "iptables -I INPUT -s \"$2\" -m state --state NEW,RELATED,ESTABLISHED -j DROP"}'
```



## 2. Web Server

web 服务器其下脚本程序的权限：400

images 等需要用户上传文件的目录权限：500

### 2.1. 目录访问控制

```
<Location ~ "/((static/)|(css/)|(images/)).*\\.php">
    Order Deny,Allow
    Deny from all
</Location>
```

### 2.2. fastcgi 安全

nginx 用户 www:www

php-fpm 用户 nobody:nogroup

### 2.3. php 函数安全

```
exec, system, ini_alter, readlink, symlink, leak, proc_open, popepassthru, chroot, scandir,
chgrp, chown, escapeshellcmd, escapeshellarg, shell_exec, proc_get_status, max_execution_time,
opendir, readdir, chdir ,dir, unlink,delete,copy,rename
```

### 2.4. 服务器版本信息

```
Apache:
ServerTokens ProductOnly
ServerSignature Off

Nginx:
server_tokens off;

PHP:
expose_php Off

Tomcat:
server="Your App Server"
```

细节参考 《Netkiller Linux 手札》

第 23 章 服务器安全

[起始页](#)

第 24 章 运维自动化



[Home](#) | [Mirror](#) | [Search](#)



# 第 24 章 运维自动化

## 目录

- [1. Cobbler](#)
- [2. Cfengine](#)
- [3. Puppet](#)
- [4. func](#)

## 1. Cobbler

<http://cobbler.github.com/>



## 2. Cfengine

<http://www.cfengine.org/>



# 3. Puppet

http://www.puppetlabs.com



# 4. func

https://fedorahosted.org/func/



# 部分 IV. SQA (Software Quality Assurance)

目录

[25. SCM \(Software Configuration Management\)](#)

[26. Automatic Testing](#)

- [1. 静态内容并发测试](#)
- [2. 动态内容测试](#)
- [3. 业务逻辑并发能力](#)
- [4. Cluster Testing](#)

[27. Software Testing](#)

- [1. 缺陷管理工具](#)
  - [1.1. Mantis](#)
    - [1.1.1. JpGraph](#)
  - [1.2. Bugzilla](#)

[28. IO Testing](#)

- [1. fio](#)
- [2. iozone](#)
- [3. iometer](#)
- [4. 测试文件系统](#)
- [5. iperf: Tool for measuring TCP and UDP bandwidth performance](#)

[29. 压力测试工具](#)

- [1. TCP/IP 压力测试优化](#)
- [2. ApacheBench](#)
- [3. JMeter](#)
- [4. Siege](#)

[30. Test - System performance benchmark](#)

[1. SysBench](#)

[31. OpenSTA\(Open System Testing Architecture\)](#)

[32. Web Testing](#)

[1. Browser Config](#)

[1.1. Chrome](#)

[2. WATIR - WEB AUTOMATION TESTING IN RUBY](#)

[2.1. Install of WATIR](#)

[2.1.1. IE](#)

[2.1.2. Firefox](#)

[2.2. Getting Started](#)

[2.3. Config file](#)

[2.4. Watir library](#)

[2.5. Browser](#)

[2.5.1. Environment variable](#)

[2.5.2. Browser options](#)

[2.5.3. Open a browser \(default: Internet Explorer\)](#)

[2.5.4. Watir::Browser.attach](#)

[2.5.5. Open Browser at the specified URL](#)

[2.5.6. Go to a specified URL](#)

[2.5.7. Close the browser](#)

[2.6. Test Unit](#)

[2.7. HTML Tags and Watir Methods](#)

[2.7.1. Collections of HTML Elements](#)

[3. Selenium - Web Browser Automation](#)

[3.1. ruby](#)

[3.2. python](#)

[4. WEBDRIVER](#)

[5. WatiN](#)

[6. PAMIE](#)





# 第 25 章 SCM (Software Configuration Management)





## 第 26 章 Automatic Testing

### 目录

- [1. 静态内容并发测试](#)
- [2. 动态内容测试](#)
- [3. 业务逻辑并发能力](#)
- [4. Cluster Testing](#)

## 1. 静态内容并发测试

apache 自带 ab 测试命令



## 2. 动态内容测试

简单动态内容展示，可以 apache 自带 ab 测试命令



### 3. 业务逻辑并发能力

如测试一定时间内，系统能承受多少笔交易

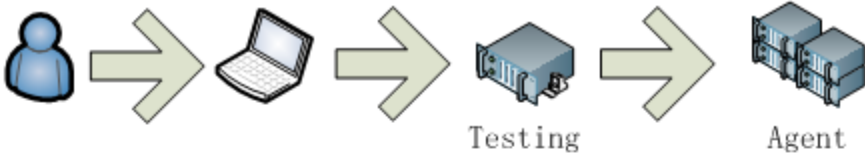
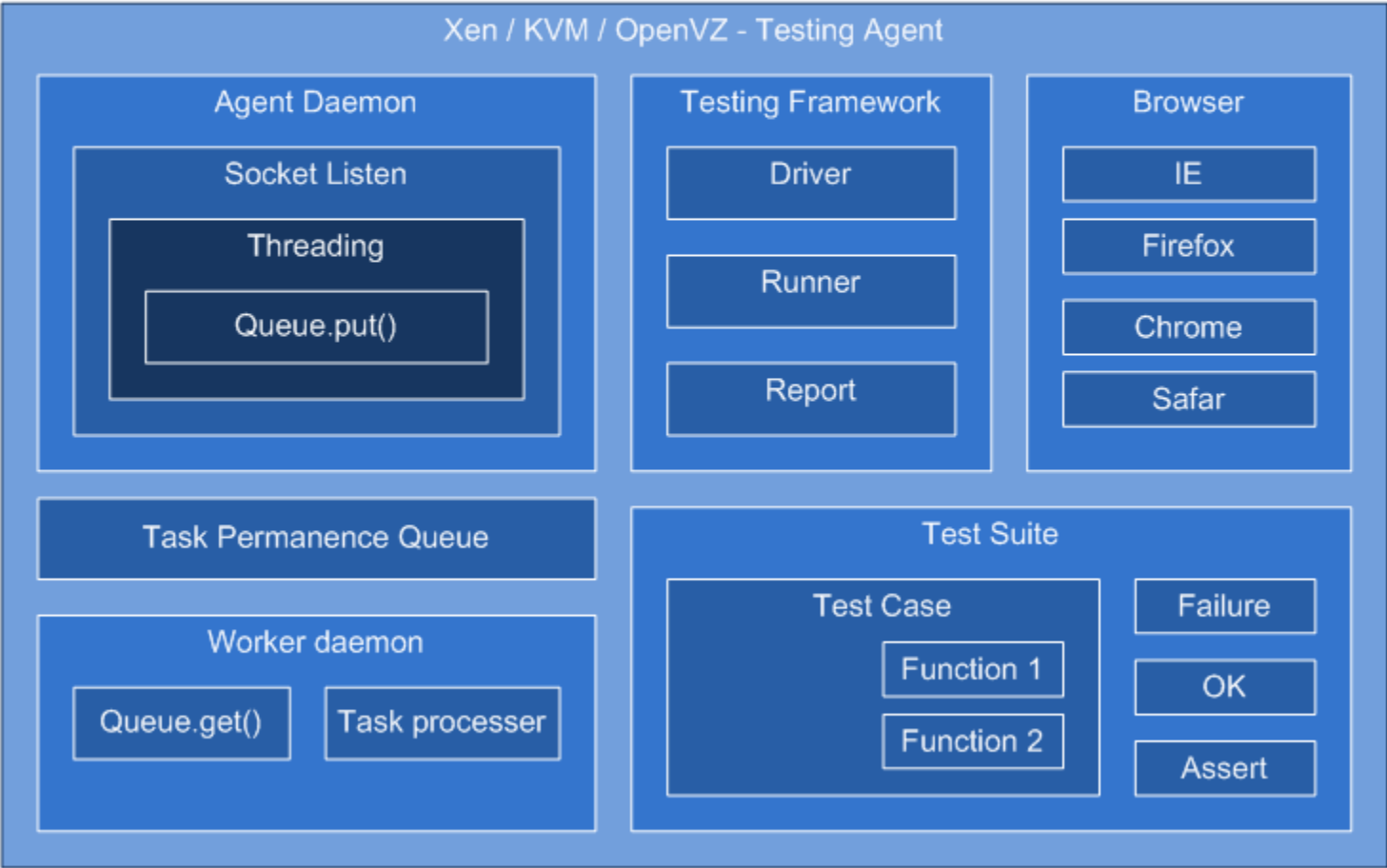
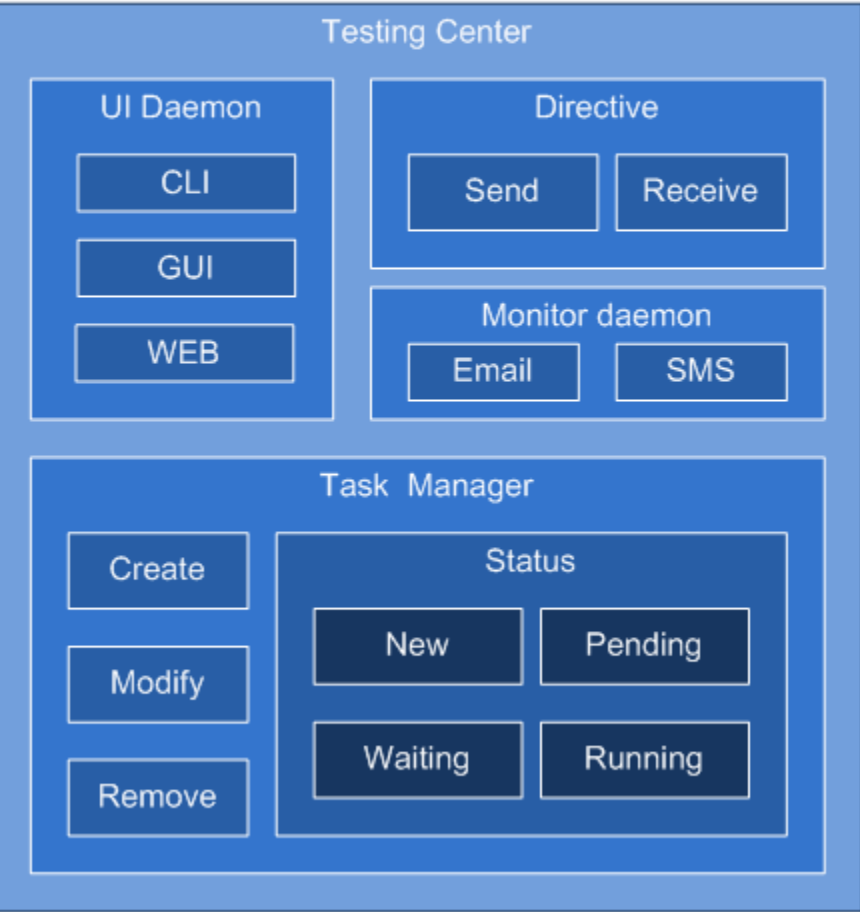
对于简单的业务逻辑测试可以使用curl模拟，在使用多线程进行压力



# 4. Cluster Testing

# Automatic Testing Cluster

Homepage: [Http://netkiller.github.com/](http://netkiller.github.com/) , [Http://netkiller.sourceforge.net/](http://netkiller.sourceforge.net/) Design by Neo



[上一页](#)

3. 业务逻辑并发能力

[上一级](#)

[起始页](#)

[下一页](#)

第 27 章 Software Testing



# 第 27 章 Software Testing

## 目录

### [1. 缺陷管理工具](#)

#### [1.1. Mantis](#)

##### [1.1.1. JpGraph](#)

#### [1.2. Bugzilla](#)

<http://openqa.org/>

[http://en.wikipedia.org/wiki/Software\\_testing](http://en.wikipedia.org/wiki/Software_testing)

# 1. 缺陷管理工具

## 1.1. Mantis

<http://www.mantisbt.org/>

### 1.1.1. JpGraph

<http://www.aditus.nu/jpgraph/index.php>

```
# wget http://hem.bredband.net/jpgraph2/jpgraph-3.0.7.tar.bz2
# tar jxvf jpgraph-3.0.7.tar.bz2
# mkdir /usr/local/share/php
# mv docportal /usr/local/share/php

# mkdir /var/www/html/mantisbt/jpgraph
# mv src/* mantisbt/jpgraph/
```

php.ini

```
# vim /etc/php.ini
include_path = ".:/usr/share/pear:/var/www/html/mantisbt/jpgraph/"
```

config\_inc.php

```
# vim config_inc.php
$g_use_jpgraph = ON;
$g_jpgraph_path = '/var/www/html/mantisbt/jpgraph/' ;
```

Don't forget to add the trailing '/'.



[Home](#) | [Mirror](#) | [Search](#)



# 第 28 章 IO Testing

## 目录

- [1. fio](#)
- [2. iozone](#)
- [3. iometer](#)
- [4. 测试文件系统](#)
- [5. iperf: Tool for measuring TCP and UDP bandwidth performance](#)

测试工具： IOMeter、 IOZone、 Sanergy、 StorNex

## 1. fio

```
$ sudo apt-get install fio
```

```
$ cat test.fio
[random-rw]
rw=randrw
size=256m
directory=/tmp/fio/data

[write]
rw=write
size=2048m
directory=/tmp/fio/data

[read]
rw=read
size=2048m
directory=/tmp/fio/data
```

```
$ fio test.fio
```

### threads

```
$ cat four-threads-randio.fio
; Four threads, two query, two writers.

[global]
rw=randread
size=256m
directory=/tmp/fio-testing/data
ioengine=libaio
iodepth=4
invalidate=1
direct=1

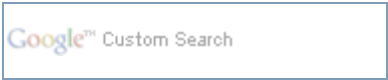
[bgwriter]
rw=randwrite
iodepth=32

[queryA]
iodepth=1
ioengine=mmap
direct=0
thinktime=3

[queryB]
iodepth=1
ioengine=mmap
direct=0
thinktime=5
```



[bgupdater]  
rw=randrw  
iodepth=16  
thinktime=40  
size=32m



## 2. iozone

http://www.iozone.org/

```
$ apt-cache search iozone
iozone3 - Filesystem and Disk Benchmarking Tool

$ sudo apt-get install iozone3

$ iozone -a
```



# 3. iometer

<http://www.iometer.org/>



## 4. 测试文件系统

测试目录深度

```
for dir in {1..66000}; do mkdir $dir; done
```

测试创建文件数

```
for file in {1..100000}; do touch file.$file; done
```



5. iperf: Tool for measuring TCP and UDP bandwidth performance



# 第 29 章 压力测试工具

## 目录

[1. TCP/IP 压力测试优化](#)

[2. ApacheBench](#)

[3. JMeter](#)

[4. Siege](#)

## 1. TCP/IP 压力测试优化

```
cat >> /etc/security/limits.conf <<EOF
root      soft  nofile 40960
root      hard  nofile 65536
nobody    soft  nofile 40960
nobody    hard  nofile 65536
nginx     soft  nofile 40960
nginx     hard  nofile 65536
EOF

cat >> /etc/security/limits.conf <<EOF
*          soft  nofile 1050000
*          hard  nofile 1050000
EOF

cat >> /etc/sysctl.conf <<EOF
net.core.rmem_default = 10000000
net.core.rmem_max = 32949076
net.core.wmem_default = 10000000
net.core.wmem_max = 32949076
net.core.optmem_max = 32949076
net.core.somaxconn = 1024

net.ipv4.ip_local_port_range = 1024 65500
net.ipv4.tcp_sack = 0
net.ipv4.tcp_syncookies = 0
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_keepalive_time = 1200
net.ipv4.tcp_max_syn_backlog = 8192
net.ipv4.tcp_max_tw_buckets = 10240
EOF
/sbin/sysctl -p
```

net.ipv4.tcp\_syncookies = 0必须禁用

net.ipv4.tcp\_max\_tw\_buckets 尽可能大，增加TIME\_WAIT列队长度

如果是WEB服务器,开启状态监视

```
location /nginx_status {
    stub_status on;
    access_log on;
}
```

测试客户端

```
cat >> /etc/security/limits.conf <<EOF
nginx    soft nofile 1000000
nginx    hard nofile 1000000
EOF

or

ulimit -SHn 1090000
```

观察TCP状态

```
# cat tcp.sh
netstat -n | awk '/^tcp/ {++state[$NF]} END {for(key in state) print key,"\t",state[key]}'

# watch ./tcp.sh
```



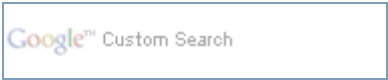
## 2. ApacheBench

```
ab -n 20000 -c 20000 http://172.16.0.8/  
ab -n 2000000 -c 20000 -t 30 -H 'Accept-Encoding: gzip' -k http://172.16.0.2/
```

循环请求

```
while true; do ab -n 10000 -c 10000 -t 30 -H 'Accept-Encoding: gzip' -k http://172.16.0.2/ &&  
sleep 5; done
```





### 3. JMeter



# 4. Siege

http://www.joedog.org/index/siege-home

Debian/Ubuntu

```
neo@debian:~$ apt-cache search Siege
siege - Http regression testing and benchmarking utility

neo@debian:~$ sudo apt-get install siege
```

.siegerc

```
verbose = false
internet = true
benchmark = true
```

siege参数介绍

## 参数

- -c, --concurrent=NUM 并发用户量，默认为10
- -u, --url="URL"
- -i, --internet INTERNET user simulation, hits the URLs randomly.
- -b, --benchmark 基准测试, signifies no delay for time testing.
- -t, --time=NUMm 持续时间，单位默认为分"m" 同样可以使用 S, M, 或 H 例如: --time=1H, 一小时.
- -r, --reps=NUM REPS, number of times to run the test, default is 25
- -f, --file=FILE FILE, url列表配置文件.

## 参数

- Transactions 总共完成了X次请求
- Availability 有效性
- Elapsed time 花费时间
- Data transferred传输数据X兆
- Response time 响应时间

- Transaction rate每秒传输率：次/每秒
- Throughput 每秒数据吞吐量
- Concurrency 实际并发
- Successful transactions 成功的传输
- Failed transactions 失败的传输
- Longest transaction 传输所花最长时间
- Shortest transaction 传输所花最短时间

```
siege -c 1000 -d 30 http://172.16.0.8/

siege -c 1000 -r 50 -i -b -t 2 -d 30 -f test.txt

cat test.txt
http://172.16.0.2/index.html
http://172.16.0.2/list.html
http://172.16.0.2/product.html
http://172.16.0.2/login.html
```



# 第 30 章 Test - System performance benchmark

目录

[1. SysBench](#)

## 1. SysBench

处理器测试

```
sysbench --test=cpu --cpu-max-prime=20000 run
多线程测试
sysbench --test=threads --num-threads=64 --thread-yields=100 --thread-locks=2 run
```

磁盘IO性能测试

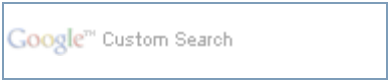
```
sysbench --test=fileio --num-threads=16 --file-total-size=3G --file-test-mode=rndrw prepare
sysbench --test=fileio --num-threads=16 --file-total-size=3G --file-test-mode=rndrw run
sysbench --test=fileio --num-threads=16 --file-total-size=3G --file-test-mode=rndrw cleanup
```

内存测试

```
sysbench --test=memory --memory-block-size=8k --memory-total-size=4G run
```

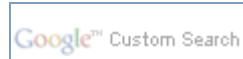
MySQL OLTP 事务处理性能测试

```
sysbench --test=oltp --mysql-table-engine=myisam --oltp-table-size=1000000 --mysql-socket=/tmp/mysql.sock --mysql-user=test --mysql-host=localhost --mysql-password=test prepare
sysbench --num-threads=8 --max-requests=1000000 --test=oltp --mysql-table-engine=innodb --oltp-table-size=10000000 --mysql-socket=/tmp/mysql.sock --mysql-user=root --mysql-host=localhost --mysql-password=xxxx --mysql-db=sbtest run
```



# 第 31 章 OpenSTA(Open System Testing Architecture)

[Home](#) | [Mirror](#) | [Search](#)



## 第 32 章 Web Testing

### 目录

#### [1. Browser Config](#)

##### [1.1. Chrome](#)

#### [2. WATIR - WEB AUTOMATION TESTING IN RUBY](#)

##### [2.1. Install of WATIR](#)

###### [2.1.1. IE](#)

###### [2.1.2. Firefox](#)

##### [2.2. Getting Started](#)

##### [2.3. Config file](#)

##### [2.4. Watir library](#)

##### [2.5. Browser](#)

###### [2.5.1. Environment variable](#)

###### [2.5.2. Browser options](#)

###### [2.5.3. Open a browser \(default: Internet Explorer\)](#)

###### [2.5.4. Watir::Browser.attach](#)

###### [2.5.5. Open Browser at the specified URL](#)

###### [2.5.6. Go to a specified URL](#)

###### [2.5.7. Close the browser](#)

##### [2.6. Test Unit](#)

##### [2.7. HTML Tags and Watir Methods](#)

###### [2.7.1. Collections of HTML Elements](#)

#### [3. Selenium - Web Browser Automation](#)

##### [3.1. ruby](#)

##### [3.2. python](#)

4. WEBDRIVER

5. WatiN

6. PAMIE

# 1. Browser Config

## 1.1. Chrome

about:flags



## 2. WATIR - WEB AUTOMATION TESTING IN RUBY

### 2.1. Install of WATIR

### 2.1. Install of WATIR

```
sudo apt-get install ruby      # for Ubuntu / Debian users
sudo yum install ruby          # for Red Hat / Fedora users
```

#### Update RubyGems

```
sudo apt-get install rubygems # for Ubuntu / Debian users
sudo yum install rubygems     # for Red Hat / Fedora users
```

#### Install Watir

```
sudo gem install firewatir

Successfully installed firewatir-1.8.0
1 gem installed
Installing ri documentation for firewatir-1.8.0...
Installing RDoc documentation for firewatir-1.8.0...
```

#### 2.1.1. IE

<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=95e06cbe-4940-4218-b75d-b8856fced535>

#### 2.1.2. Firefox

#### JSSH for Firefox:

<http://wiki.openqa.org/display/WTR/FireWatir+Installation>

test jssh

```
$ firefox -jssh
$ telnet localhost 9997
```

## 2.2. Getting Started

### Quick Start

create a testing script

```
$ cat google.rb
require 'rubygems'
require 'watir'
require 'firewatir'
```



```
browser = Watir::Browser.new
browser.goto 'http://google.com'
browser.text_field(:name, 'q').set 'Ruby'
browser.button(:name, 'btnG').click
if browser.text.include? 'Ruby Programming Language'
  puts 'Text was found'
else
  puts 'Text was not found'
end
```

run test

```
$ irb google.rb

google.rb(main):001:0> require 'rubygems'
=> true
google.rb(main):002:0> require 'watir'
=> true
google.rb(main):003:0> require 'firewatir'
=> true
google.rb(main):004:0>
google.rb(main):005:0* browser = Watir::Browser.new
=> #<FireWatir::Firefox:0x7f797dd2b880 url="http://www.google.com.hk/search?hl=zh-CN&source=hp&biw=1440&bih=660&q=Ruby&btnG=Google+%E6%90%9C%E7%B4%A2&aq=f&aqi=g10&aql=&oq=" title="Ruby - Google \346\220\234\347\264\242">
google.rb(main):006:0> browser.goto 'http://google.com'
=> #<FireWatir::Firefox:0x7f797dd2b880 url="http://www.google.com.hk/" title="Google">
google.rb(main):007:0> browser.text_field(:name, 'q').set 'Ruby'
=> 0
google.rb(main):008:0> browser.button(:name, 'btnG').click
=> 0
google.rb(main):009:0> if browser.text.include? 'Ruby Programming Language'
google.rb(main):010:1>   puts 'Text was found'
google.rb(main):011:1> else
google.rb(main):012:1*   puts 'Text was not found'
google.rb(main):013:1> end
Text was found
=> nil
```

### 2.3. Config file

Create a file like this, called options.yml

```
cat options.yml
browser: ie
```

In your program, execute the following code before calling "Watir::Browser.new":

```
Watir.options_file = '~/options.yml'
```

You will need to provide the full path to your file. Once you have done this, "Browser.new" will use the option settings from this file. If you also use an environment variable, it will override this setting.

Currently the following options are configurable for IE:

```
speed: fast
visible: true
```

Valid speed settings are "fast", "slow" and "zippy"

### 2.4. Watir library

Load the Watir library

```
require 'rubygems'
require 'watir'
require 'firewatir'
```

### 2.5. Browser

### 2.5.1. Environment variable

```
> set watir_browser=ie
> set watir_browser=firefox
> set watir_browser=safari
```

### 2.5.2. Browser options

```
Watir::Browser.default = 'ie'
Watir::Browser.default = 'firefox'
```

#### Specifying additional options (IE only)

Speed up execution  
(or use the "-b" command line switch)

```
browser.speed = :fast
Maximize browser window
```

```
browser.maximize
Pop browser window to front
```

```
browser.bring_to_front
```

### 2.5.3. Open a browser (default: Internet Explorer)

```
browser = Watir::Browser.new
```

### 2.5.4. Watir::Browser.attach

```
$ firefox -jssh
```

```
browser = Watir::Browser.attach(:title, 'Mozilla Firefox')
```

### 2.5.5. Open Browser at the specified URL

```
browser = Watir::Browser.start("http://google.com")
```

### 2.5.6. Go to a specified URL

```
browser.goto("http://amazon.com")
```

### 2.5.7. Close the browser

```
browser.close
```

## 2.6. Test Unit

#### Create a Test Instance

```
require 'test/unit'
require 'rubygems'
require 'watir'
require 'firewatir'

class GoogleHomePage < Test::Unit::TestCase
  def test_there_should_be_text_About_Google
    browser = Watir::Browser.start "http://www.google.com"
    assert(browser.text.include?("About Google"))
  end
end
```

```
$ ruby test_unit.rb

Loaded suite test_unit
Started
.
Finished in 1.687 seconds.

1 tests, 1 assertions, 0 failures, 0 errors
```

Create a Test Case Method

```
class SampleTest < Watir::TestCase
  def test_login
    # login test code, etc
  end

  def test_account
    # account test code, etc
  end
end
```

How to run a specific test case?

```
ruby my_test_file.rb --name test_account
```

提示

必须test\_为前缀

默认非顺序执行

顺序执行方法 test\_a\_mytest, test\_b\_mytest; test\_01\_mytest, test\_02\_mytest

2.7. HTML Tags and Watir Methods

2.7.1. Collections of HTML Elements

If you have a page that looks like this:

```
<a href="http://google.com">google.com</a>
<a href="http://facebook.com">facebook.com</a>
You can get href attribute of all links with:
```

```
browser.links.each do |link|
  puts link.href
end
When you execute the code, output should be:

http://google.com/
http://facebook.com/
```



## 3. Selenium - Web Browser Automation

<http://selenium.seleniumhq.org/>

### 3.1. ruby

```
gem install selenium-webdriver
```

### 3.2. python

```
apt-get install python-pip
sudo pip install -U selenium

wget http://selenium.googlecode.com/files/selenium-server-standalone-2.0b3.jar
java -jar selenium-server-standalone-2.0b3.jar

vim test.py

from selenium import webdriver
from selenium.common.exceptions import NoSuchElementException
from selenium.common.keys import Keys
from time

browser = webdriver.Firefox() # Get local session of firefox
browser.get("http://www.yahoo.com") # Load page
assert browser.title == "Yahoo!"
elem = browser.find_element_by_name("p") # Find the query box
elem.send_keys("selenium" + Keys.RETURN)
time.sleep(0.2) # Let the page load, will be added to the API
try:
    browser.find_element_by_xpath("//a[contains(@href,'http://seleniumhq.org')]")
except NoSuchElementException:
    assert 0, "can't find seleniumhq"
browser.close()
```



# 4. WEBDRIVER

<http://code.google.com/p/webdriver/wiki/GettingStarted>



# 5. WatiN

http://watin.org/



## 6. PAMIE

PAMIE是一套为Python写的用于Web自动化测试的工具，采用Win32COM的方式操作IE来实现。原理比较简单，但是可以真实再现用户的使用，QA用非常合适



# 部分 V. DevOps

目录

[33. What is DevOps](#)

- [1. Trac Project Management](#)
- [2. Development Environment](#)
- [3. Testing Environment](#)
- [4. Coding workflow](#)
- [5. Automation Deployment](#)
  - [5.1. Deployment Toolkit](#)
  - [5.2. Deployment Script](#)
    - [5.2.1. FTP Script](#)
    - [5.2.2. SFTP Script](#)
    - [5.2.3. SCP Script](#)
- [6. Upgrade](#)





# 第 33 章 What is DevOps

## 目录

- [1. Trac Project Management](#)
- [2. Development Environment](#)
- [3. Testing Environment](#)
- [4. Coding workflow](#)
- [5. Automation Deployment](#)

[5.1. Deployment Toolkit](#)

[5.2. Deployment Script](#)

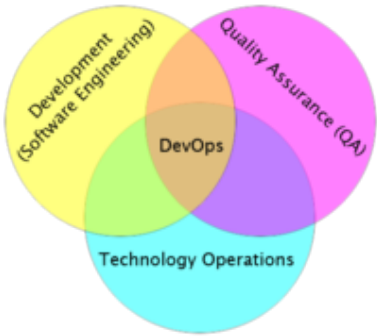
[5.2.1. FTP Script](#)

[5.2.2. SFTP Script](#)

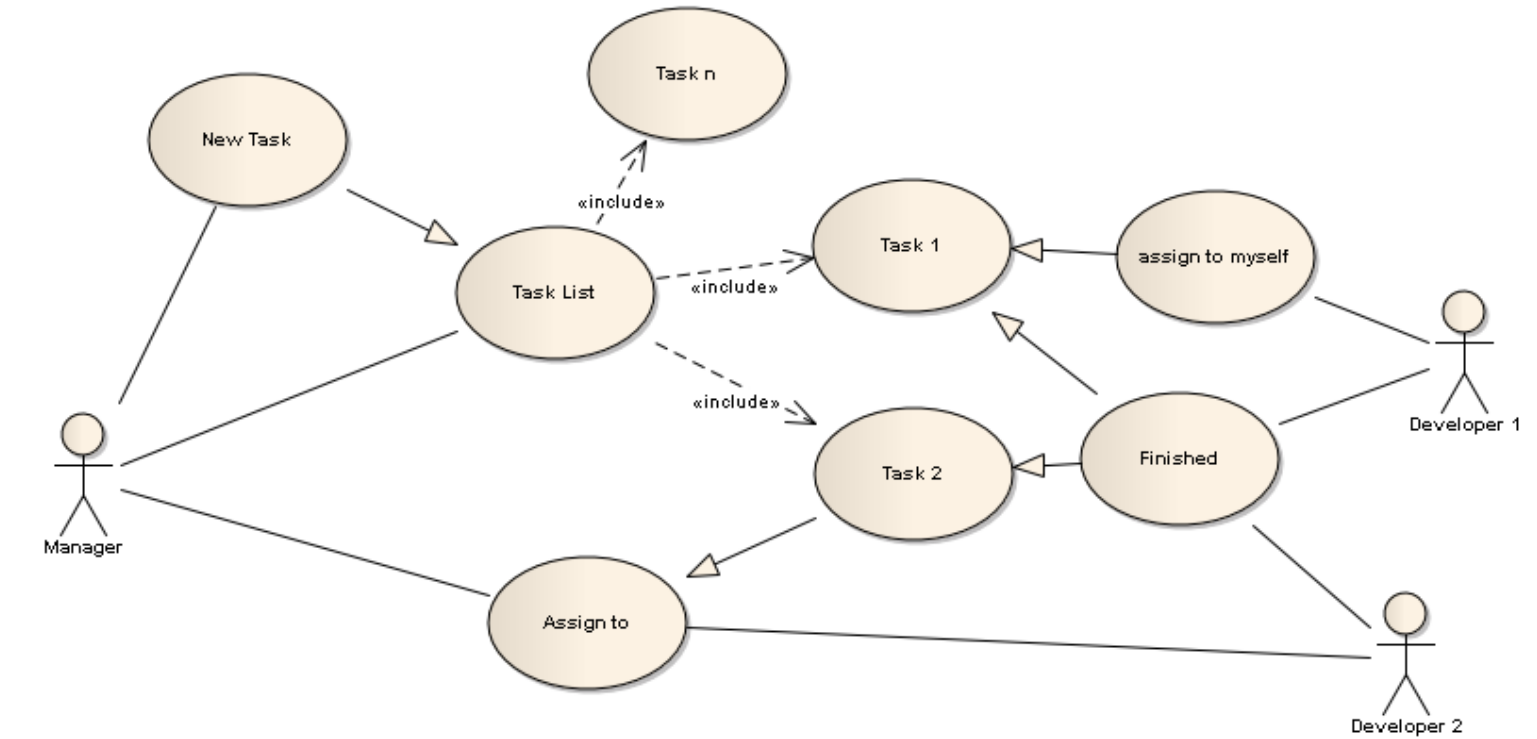
[5.2.3. SCP Script](#)

[6. Upgrade](#)

What is [DevOps](#)?

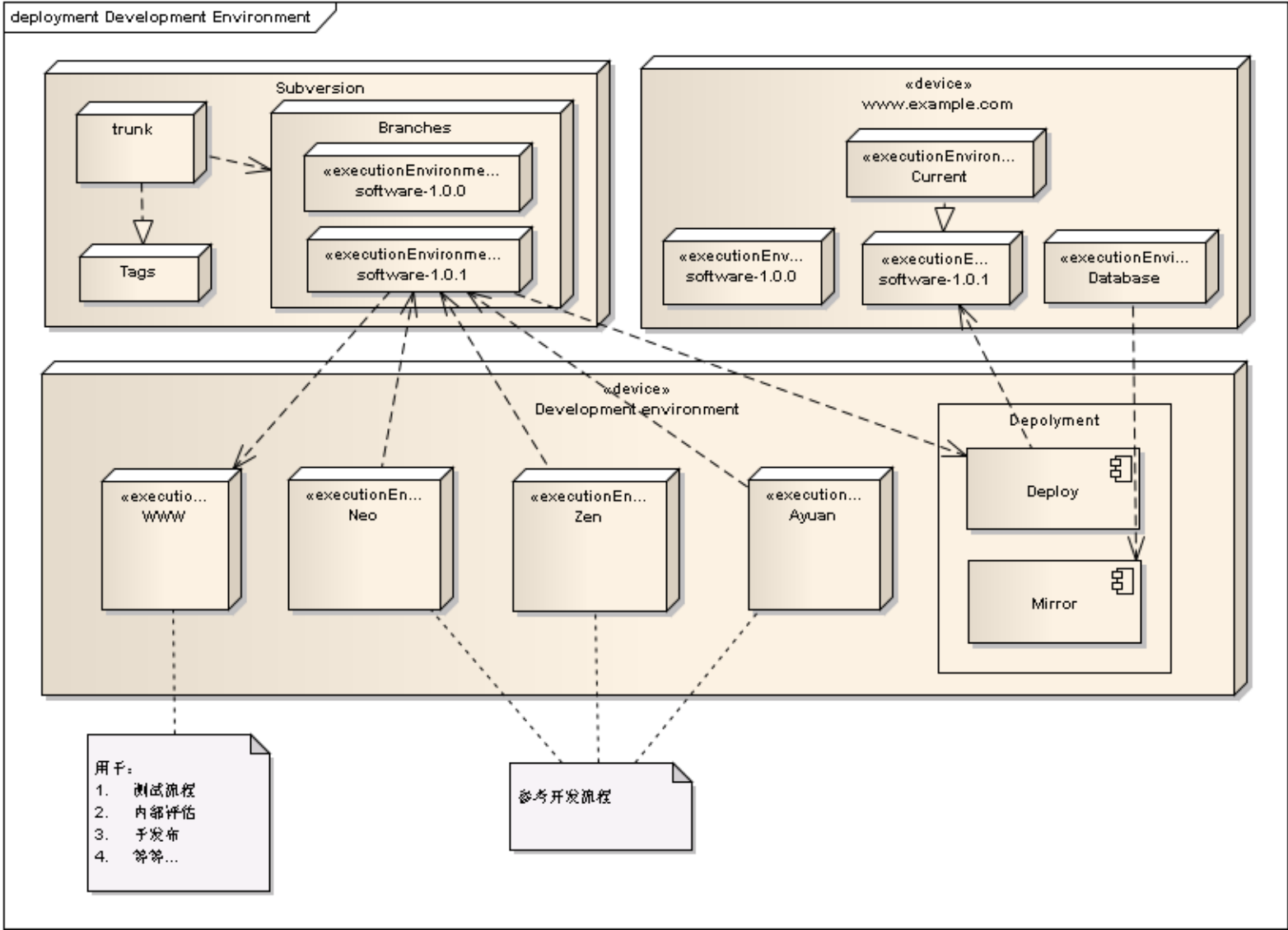


## 1. Trac Project Management



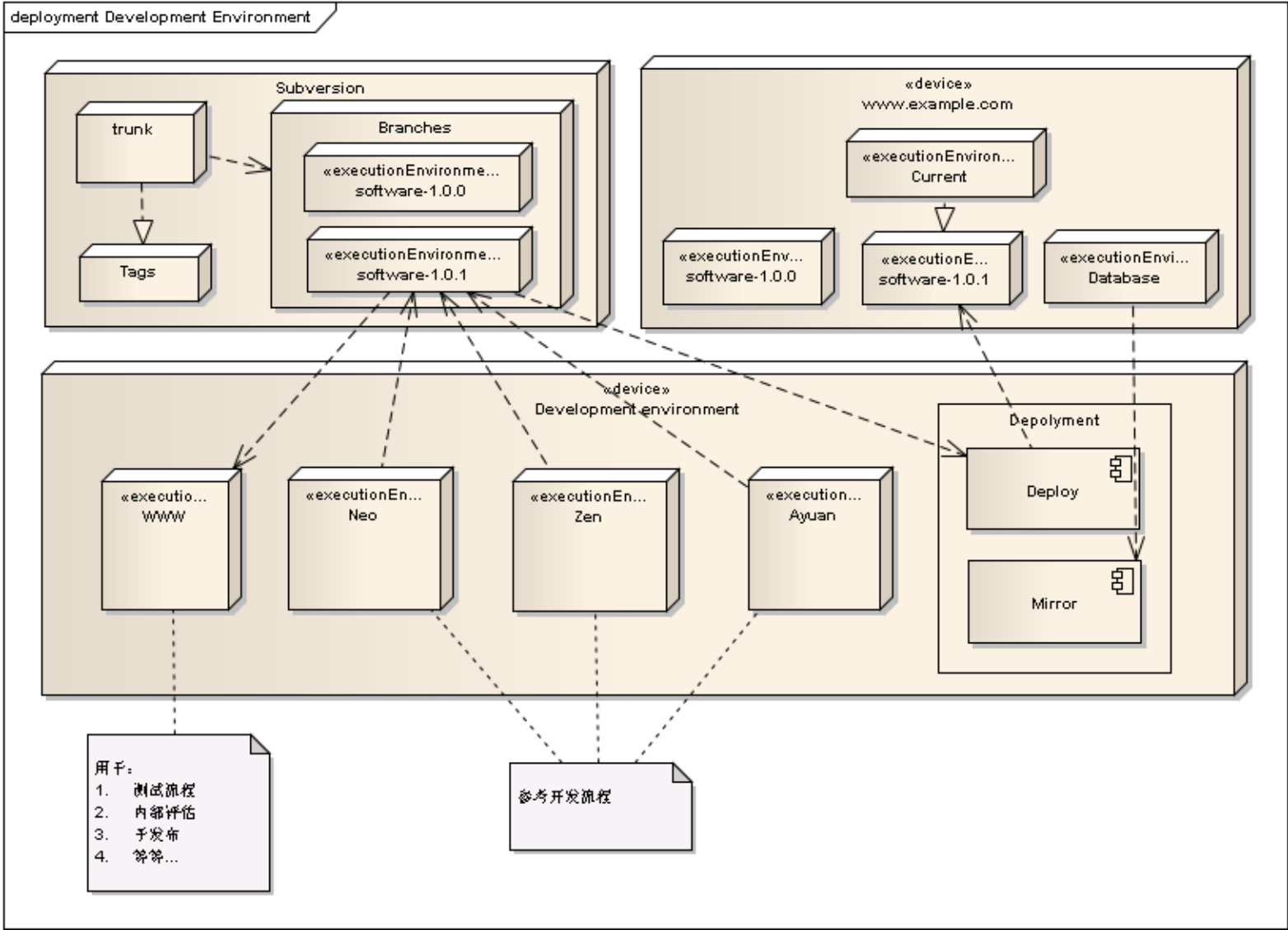


2. Development Environment



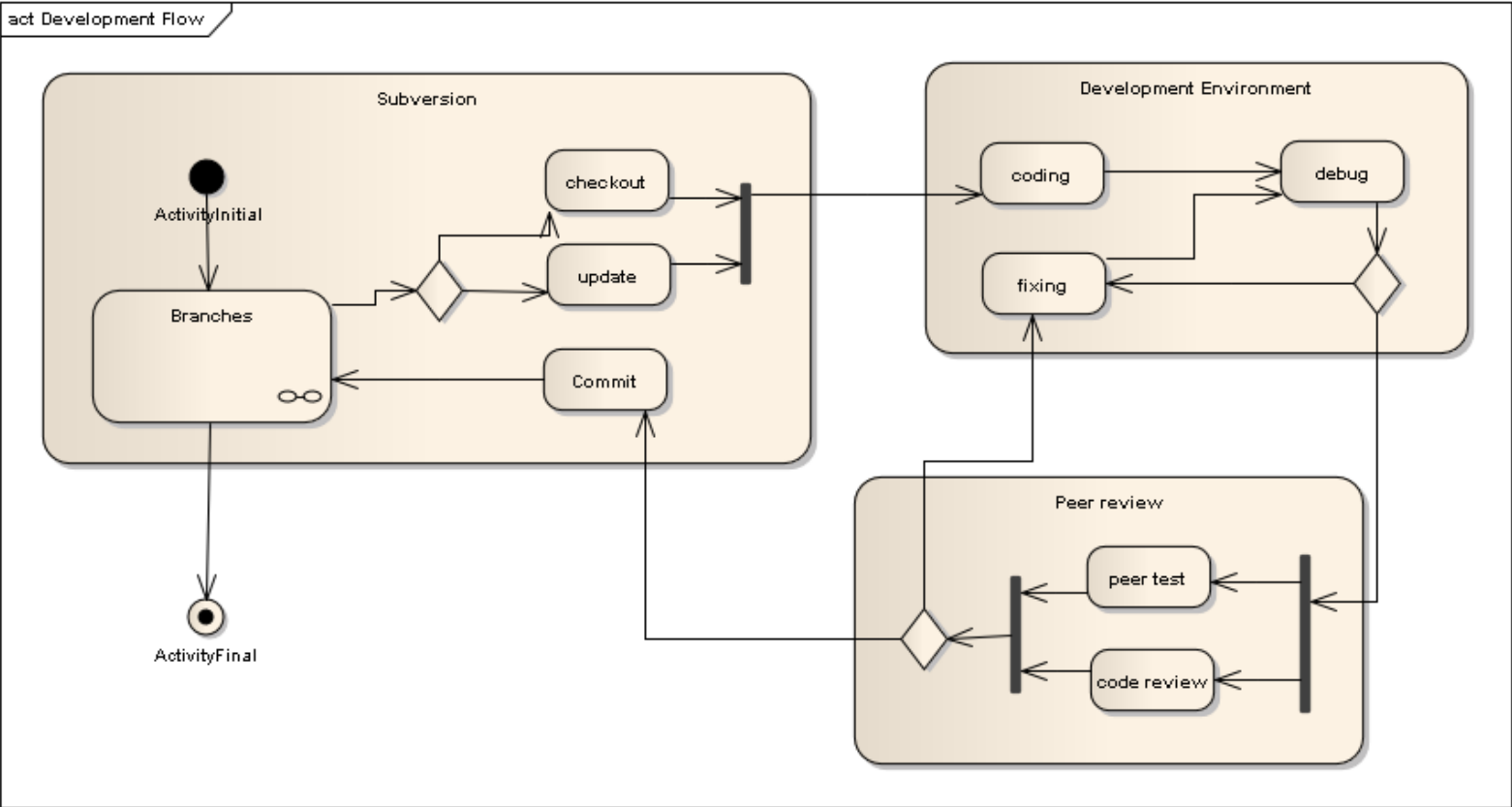


3. Testing Environment





4. Coding workflow



[Home](#) | [Mirror](#) | [Search](#)



## 5. Automation Deployment

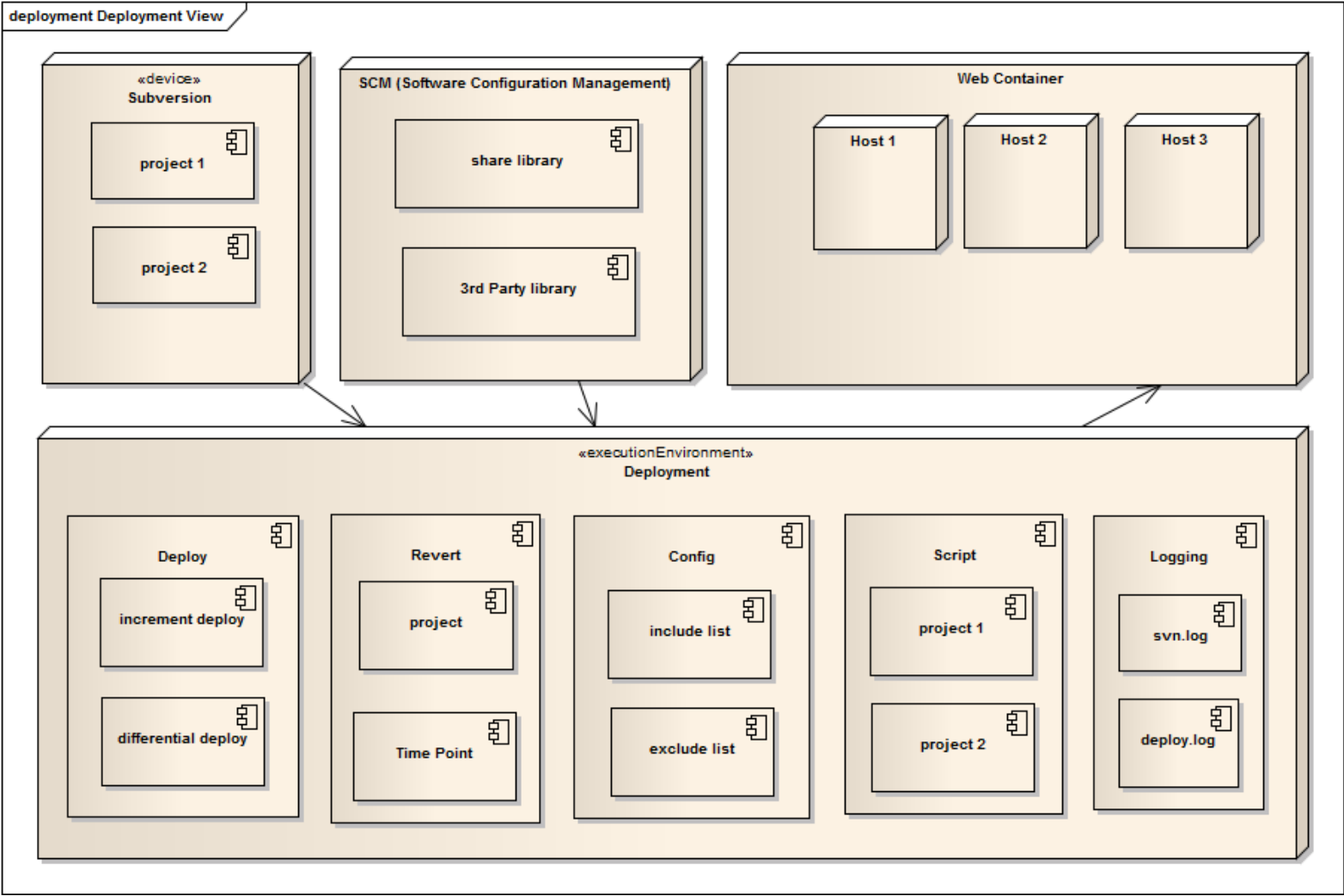
传统ftp上传已经不适合在大规模环境下应用。

另外我们不得不考虑传输过程中的安全问题。

常用部署工具

1. FileZilla Client, Winscp
2. Wget, ncftp
3. Rsync, Csync2
4. Unison

### 5.1. Deployment Toolkit



deploy

```
#!/bin/bash
. deploy.conf

#datetime=`date '+%Y%m%d-%H%M%S'`
#pkgname=${project}-${version}-${datetime}.pkg
#if [ -d /tmp/${project} ];then
#   rm -rf /tmp/${project}
#fi
#echo "clean "
#svn export ${repository} /tmp/${project} > deploy.log
```

```
#echo export svn from repository
#tar jcvf ${pkgname} /tmp/${project} --remove-files >> deploy.log
#echo packaging
#scp ${pkgname} ${destination}
#echo uploading
LOGFILE=deploy.log
RSYNC="rsync -a -u -z --progress --stats -e ssh --log-file=${LOGFILE}"
if [ ! -d ${local} ]; then
    mkdir ${local}
fi

function stable {
    local edition=$(basename $stable)
    svn export ${stable} ${src}/${edition} >> /dev/null

    # for ignore in $( cat excluded.lst ); do
    #     rm -rf ${src}/${edition}/${ignore}
    # done

    $RSYNC --exclude-from=excluded.lst ${src}/${edition} ${destination}
    ssh ${remote} < script/stable > /dev/null 2>&1
}

function unstable {
    local edition=$(basename $unstable)
    svn export ${unstable} ${src}/${edition}

    for ignore in $( cat excluded.lst ); do
        rm -rf ${src}/${edition}/${ignore}
    done

    $RSYNC ${src}/${edition} ${destination}

    ssh ${remote} < script/unstable >> ${LOGFILE}
}

function nightly {
    local edition=$(basename $unstable)
    svn export ${unstable} ${src}/${nightly}
    $RSYNC --exclude-from=exclude.conf ${src}/${nightly} ${destination}
    # ssh ${remote} < script/unstable
}

function clean {
    rm -rf ${src}
}

function switch {
    ssh $remote "rm /www/example ; ln -s /www/$1 /www/example"
}

case "$1" in
    switch)
        switch $2
        ;;
    stable)
        clean
        stable
        ;;
    unstable)
        clean
        unstable
        ;;
    nightly)
        clean
        nightly
        ;;
    clean)
        clean
        ;;
    *)
        echo $"Usage: $0 {stable|unstable|nightly|clean}"
        echo $"          $0 switch {stable|unstable|nightly}"
        exit 1
esac
```

deploy.conf

```
repository=svn://172.16.0.1/example
stable=${repository}/trunk
unstable=${repository}/branches/example-unstable-2.7.0-3
remote=rsync@www.example.com
destination=${remote}:/www
project=$(basename $repository)
src=/tmp/${project}
version=1.0
```

excluded.lst/included.lst

```
install
images/20*
images/ads
temp
data/config.php
data
themes/default/home
themes/default/block
themes/default/special/*
themes/green
static/*
```

script/stable|unstable|nightly

```
find /www/ -type d -exec chmod 755 {} \;
find /www/ -type f -exec chmod 644 {} \;
```

```
chmod 777 -R /www/project/temp > /dev/null
chmod 777 -R /www/project/data > /dev/null
chmod 777 -R /www/project/images > /dev/null
```

5.2. Deployment Script

```
$ /srv/bin/srvmgr
Usage: /srv/bin/srvmgr [OPTION] <server-id> <directory/timepoint>

OPTION:
    list
    list <name>

    conf list
    conf new <project>
    conf remove <project>
    conf show <project>
    conf edit <project>

    backup <project>
    deploy <project> <timepoint>

    import <project> <dir>
    export <project> <dir>

$ cat /srv/bin/srvmgr
#!/bin/bash
#=====
# netkiller series utilities - script by Neo <openunix@l63.com>
# http://netkiller.sourceforge.net/
# $Id$
#=====

PREFIX=/srv
BACKUP_DIR=~/.backup
TMPDIR=/tmp
#SUBVERSION=
UPLOAD_DIR=$TMPDIR/.upload
SVN=/usr/bin/svn
REPOSITORIES=http://192.168.0.1/example/deploy
#=====
BINDIR=$PREFIX/bin
SBINDIR=$PREFIX/sbin
#LIBEXECDIR=$PREFIX/libexec
#DATADIR=$PREFIX/data
SYSCONFDIR=$PREFIX/etc
#MANDIR
SRCDIR=$PREFIX/src
#=====

function usage(){
echo $"Usage: $0 [OPTION] <server-id> <directory/timepoint>"
echo -ne "
OPTION:
    list
    list <name>

    conf list
    conf new <project>
    conf remove <project>
    conf show <project>
    conf edit <project>

    backup <project>
    deploy <project> <timepoint>

    import <project> <dir>
    export <project> <dir>
"
    exit
}

function list(){
#mkdir deploy
#mkdir backup

    local prj=$2

    if [ -z $prj ]; then
        $SVN list $REPOSITORIES
    else
        $SVN list $REPOSITORIES/$prj
    fi
}

function conf(){
    local cmd=$2
    local prj=$3
    case $cmd in
        list)
            ls $SYSCONFDIR
            ;;
        new)
            read -p "Project directory: " prjdir
            if [ -z $prjdir ]; then
                exit
            fi
            if [ -f $SYSCONFDIR/$prjdir.conf ]; then
                echo "cannot create config $prjdir.conf': File exists"
                exit 1
            fi

            read -p "subversion url: $REPOSITORIES/: " svnurl
            if [ -z $svnurl ]; then
                svnurl=$REPOSITORIES
            fi
            read -p "hostname: " host
            if [ -z $host ]; then
                host="localhost"
            fi
        ;;
    esac
}
```



```

        echo "default hostname 'localhost'"
    fi
    read -p "upload mode ftp/scp/sftp/rsync: " mode
    if [ -z $mode ]; then
        mode=ftp
    else
        case $mode in
            ftp)
                mode="ftpdeploy"
                ;;
            scp)
                mode="scpdeploy"
                ;;
            sftp)
                mode="sftpdeploy"
                ;;
            rsync)
                mode="rsync"
                ;;
        esac
    fi
    read -p "Create $prjdir config? [y/n]" -n 1 key
    echo
    if [ $key = 'y' ]; then
        echo -ne "REPOSITORIES=$REPOSITORIES/$svnurl
COMMAND=$mode
HOSTNAME=$host
" >> $SYSCONFDIR/$prjdir.conf
    fi
    ;;
remove)
    if [ -f $SYSCONFDIR/$prj ]; then
        rm -rf $SYSCONFDIR/$prj
    fi
    ;;
show)
    cat $SYSCONFDIR/$prj
    ;;
edit)
    vim $SYSCONFDIR/$prj
    ;;
*)
    usage
    ;;
esac
}

function deploy(){
    local prj=$2
    local ver=$3
    if [ -z $prj ]; then
        usage
    fi
    source $SYSCONFDIR/$prj.conf
    mkdir -p $UPLOAD_DIR
    rm -rf $UPLOAD_DIR/*
    #echo $REPOSITORIES
    #echo $COMMAND
    $SVN export $REPOSITORIES/$ver/ $UPLOAD_DIR/$ver
    $BINDIR/$COMMAND deploy $HOSTNAME $UPLOAD_DIR/$ver/
}

function import(){
    local dir=$3
    local prj=$2
    if [ -z $dir ]; then
        usage
    fi
    if [ -z $prj ]; then
        usage
    fi

    $SVN import $dir $REPOSITORIES/$prj
}

function export(){
    local prj=$2
    local dir=$3

    if [ -z $dir ]; then
        usage
    fi
    if [ -z $prj ]; then
        usage
    fi

    $SVN export $REPOSITORIES/$prj $dir
}

case "$1" in
    list)
        list $@
        ;;
    conf)
        conf $@
        ;;
    deploy)
        deploy $@
        ;;
    withdraw)
        withdraw $2 $3
        ;;
    history)
        ls -l ~/backup
        ;;
    import)
        import $@
        ;;
    export)
        export $@
        ;;
    *)
        usage
        exit 1
esac

```

```
#!/bin/bash
#=====
# http://netkiller.sourceforge.net/
# Author: neo chen
#=====
BACKUP_DIR=~/.backup
UPLOAD_DIR=~

deploy(){

    local SERVER_ID=$1
    local FILEDIR=$UPLOAD_DIR/$2
    #local filelist=$(find $FILEDIR -type f | sed -e "s:^(FILEDIR):g" | tr "\n" " ")
    #local filelist=$(find $FILEDIR -type f | sed -e "s:^(FILEDIR):g")
    cd ~
    local TIMEPOINT=$(date '+%Y-%m-%d@%H:%M:%S')
    mkdir $BACKUP_DIR/$SERVER_ID/$TIMEPOINT
    cd $BACKUP_DIR/$SERVER_ID/$TIMEPOINT

    for dir in $(find $FILEDIR -type d | sed -e "s:^(FILEDIR):g")
    do
        mkdir -p $dir
    done

    #echo $filelist > $BACKUP_DIR/$TIMEPOINT/file.lst
    echo ===== backup timepont $TIMEPOINT =====

    for file in $(find $FILEDIR -type f | sed -e "s:^(FILEDIR):g")
    do
        local lfile=$(echo $file | sed "s:^/:g")
        echo "get -z $file $lfile"
    done | ncftp $SERVER_ID

    #echo "get $filelist" | ncftp $SERVER_ID
    #echo "I have finished backup files, I will be uploading files to $SERVER_ID"
    read -p "Are you ready? [y/n]" -n 1 key
    echo
    if [ $key = 'y' ]; then
        echo ===== upload =====
        echo "put -R $FILEDIR*" | ncftp $SERVER_ID
    fi
    cd ~
    echo
}

function withdraw(){
    local SERVER_ID=$1
    local TIMEPOINT=$2
    cd $BACKUP_DIR/$SERVER_ID/$TIMEPOINT
    #local filelist=$(find -type f | sed -e "s/^\./g" | tr "\n" " ")
    echo "put -R $BACKUP_DIR/$SERVER_ID/$TIMEPOINT/*" | ncftp $SERVER_ID
    cd ~
}

usage(){
    echo $"Usage: $0 {deploy|withdraw|history} <server-id> <directory/timepoint>"
    exit 1
}

case "$1" in
    deploy)
        if [ -z $2 ]; then
            usage
            exit
        fi
        if [ -z $3 ]; then
            usage
            exit
        fi

        if [ $# -ne 2 ] ; then
            usage
            exit 2
        fi

        deploy $2 $3
        ;;
    withdraw)
        if [ -z $2 ]; then
            usage
            exit
        fi
        if [ -z $3 ]; then
            usage
            exit
        fi

        withdraw $2 $3
        ;;
    history)
        ls -l ~/.backup/$2
        ;;
    *)
        usage
        exit 1
esac
```

5.2.2. SFTP Script

```
#!/bin/bash
BACKUP_DIR=/tmp/backup
REMOTE_DIR=/tmp
UPLOAD_DIR=' '
LOCAL_DIR=' '
#LOCAL_DIR=$UPLOAD_DIR

deploy(){
    local host=$1
    SCP=${host}:$REMOTE_DIR
    TIMEPOINT=$(date '+%Y-%m-%d,%H:%M:%S')
    BACKUP_DIR=$BACKUP_DIR/${host}/${TIMEPOINT}

#    if [ ! -e $i.log ]; then
```

```
mkdir -p $BACKUP_DIR
#
fi

for dir in $(find $LOCAL_DIR -type d | sed -e "s:^$LOCAL_DIR::g");
do
    mkdir -p $BACKUP_DIR/$dir
done

echo ===== backup timepont $TIMEPOINT =====

#
#   for file in $(find $LOCAL_DIR -type f | sed -e "s:^$LOCAL_DIR::g");
#   do
#       echo $file
#       scp $SCP/$file $BACKUP_DIR/$file
#   done

for file in $(find $LOCAL_DIR -type f | sed -e "s:^$LOCAL_DIR::g")
do
    local lfile=$BACKUP_DIR/$(echo $file | sed "s:^/::g")
    echo "get $file $lfile"
done | sftp $SCP

read -p "Are you ready? [y/n]" -n 1 key ; echo
if [ $key = 'y' ]; then
    echo ===== upload =====
    eval scp -r "$LOCAL_DIR*" $SCP
fi
}
withdraw(){
    local host=$1
    local timepoint=$2
    SCP=${host}:$REMOTE_DIR
    eval scp -r $BACKUP_DIR/${host}/${timepoint}/* $SCP
}
restart(){
    /usr/local/resin-pro-3.1.3/bin/httpd.sh -server payservice restart
}

history(){
    if [ -z $1 ]; then
        usage
        exit
    else
        ls -l $BACKUP_DIR/$1
    fi
}

usage(){
    echo $"Usage: $0 {deploy|withdraw|history} <server-id> <directory/timepoint>"
    exit 1
}

case "$1" in
    deploy)

        if [ $# -ne 3 ] ; then
            usage
            exit 2
        fi

        export UPLOAD_DIR=$3
        export LOCAL_DIR=$UPLOAD_DIR

        deploy $2
        ;;
    withdraw)
        if [ $# -ne 3 ] ; then
            usage
            exit 2
        fi

        withdraw $2 $3
        ;;
    history)
        history $2
        ;;
    *)
        usage
        exit 1
esac
```

5.2.3. SCP Script

scpdeploy

```
#!/bin/bash
BACKUP_DIR=~/.backup/java
UPLOAD_DIR=.
#IP=""
#FILEDIR=$UPLOAD_DIR/"

deploy(){
    SCP=root@${IP}:/www
    TIMEPOINT=$(date '+%Y-%m-%d,%H:%M:%S')
    BACKUP_DIR=$BACKUP_DIR/$TIMEPOINT
    mkdir $BACKUP_DIR

    for dir in $(find $FILEDIR -type d | sed -e "s:^$FILEDIR::g");
    do
        #
        echo $dir
        mkdir -p $BACKUP_DIR/$dir
        #
        if [ ! -e $i.log ]; then
            #
            mkdir -p $
            #
            fi
        done

        echo ===== backup timepont $TIMEPOINT =====

        for file in $(find $FILEDIR -type f | sed -e "s:^$FILEDIR::g");
        do
```

```
#         echo $file
        scp $SCP/$file $BACKUP_DIR/$file
    done

    read -p "Are you ready? [y/n]" -n 1 key
    echo
    if [ $key = 'y' ]; then
        echo ===== upload =====

        eval scp -r "$FILEDIR*" $SCP
    fi
}

withdraw(){
    host=$1
    timepoint=$2
    SCP=root@${host}:/www
    eval scp -r $BACKUP_DIR/$timepoint/* $SCP
}

restart(){
    /usr/local/resin-pro-3.1.3/bin/httpd.sh -server payservice restart
}

usage(){
    echo $"Usage: $0 {deploy|withdraw|history} <server-id> <directory/timepoint>"
    exit 1
}

case "$1" in
    deploy)
        if [ -z $2 ]; then
            usage
            exit
        else
            export IP=$2
        fi
        if [ -z $3 ]; then
            usage
            exit
        else
            export FILEDIR=$UPLOAD_DIR/$3
        fi

        deploy
        ;;
    withdraw)
        if [ -z $2 ]; then
            usage
            exit
        fi
        if [ -z $3 ]; then
            usage
            exit
        fi

        withdraw $2 $3
        ;;
    history)
        ls -l $BACKUP_DIR
        ;;
    *)
        usage
        exit 1
esac
```



# 6. Upgrade



# 附录 A. 附录



## 术语表

SNMP

ANSI

(American National Standards Institute) This group is the U.S. member organization that belongs to the ISO, the International Organization for Standardization.

CSS

CSS是Cascading style Sheets的简称，中文译作“层叠样式表单”，在主页制作时采用CSS技术，可以有效地对页面的布局、字体、颜色、背景和其它效果实现更加精确的控制。

集群(Cluster)

所谓集群是指一组独立的计算机系统构成的一个松耦合的多处理器系统，它们之间通过网络实现进程间的通信。应用程序可以通过网络共享内存进行消息传送，实现分布式计算机。

负载均衡(Load Balancing)

网络的负载均衡是一种动态均衡技术，通过一些工具实时地分析数据包，掌握网络中的数据流量状况，把任务合理均衡地分配出去。这种技术基于现有网络结构，提供了一种扩展服务器带宽和增加服务器吞吐量的廉价有效的方法，加强了网络数据处理能力，提高了网络的灵活性和可用性。



## 2. 目录文件与权限

### 2.1. 读写权限

Apache进程所有者: nobody

程序所有者: www

apache 可以读取程序并运行，但apache 无法改写代码，/tmp等特殊目录可以写入操作

#### 重置权限命令

```
chown www:www -R /www
chown nobody:nobody -R /www/www.example.com/tmp

find /www/ -type d -exec chmod 755 {} \;
find /www/ -type f -exec chmod 644 {} \;
chmod 744 -R /www/www.example.com/tmp
```

### 2.2. 访问权限

#### 屏蔽访问权限

```
<Directory>
<DirectoryMatch>
<Files>
<FilesMatch>
<Location>
<LocationMatch>
```

并不是所有目录和文件都需要提供给用户的，例如早期PHP项目中没有使用框架，常常有include, config等等目录需要屏蔽

#### 例 12.1. Example for ECSHOP

```
<VirtualHost *:80>
    ServerAdmin webmaster@example.com
    DocumentRoot /www/www.example.com/
    ServerName www.example.com
    ServerAlias example.com
    DirectoryIndex index.html index.php
    CustomLog "|/srv/httpd/bin/rotatelog" /www/logs/www.example.com/access.%Y-%m-%d.log 86400 480"
    combined

    <Location /data/>
        Order allow,deny
        Deny from all
    </Location>
    <Location /images/upload/>
        Order allow,deny
        Deny from all
    </Location>
    <Location /temp/>
        Order allow,deny
        Deny from all
    </Location>
    <Location /includes/>
```



```
        Order allow,deny
        Deny from all
    </Location>
    <Location /library/>
        Order allow,deny
        Deny from all
    </Location>
    <Location /plugin/>
        Order allow,deny
        Deny from all
    </Location>

    <Directory /www/www.example.com/images/>
        <Files *.php>
            Order allow,deny
            Deny from all
        </Files>
    </Directory>
    <Directory /www/www.example.com/js/>
        <Files *.php>
            Order allow,deny
            Deny from all
        </Files>
    </Directory>

    <Directory /www/www.example.com/themes/>
        <Files *.php>
            Order allow,deny
            Deny from all
        </Files>
    </Directory>
</VirtualHost>
```

[Home](#) | [Mirror](#) | [Search](#)



# 第 11 章 性能优化

## 目录

### [1. 尽量使用单引号](#)

#### 1. 尽量使用单引号

尽量使用单引号，迫不得已才使用双引号，因为双引号会处理转义字符。



7. 环境安全

7.1. 开发环境

7.2. 测试环境

7.3. 程序

7.4. 数据库



# 第 19 章 Linux 数据备份

## 目录

- [1. 批量差异备份](#)
- [2. tar 备份](#)

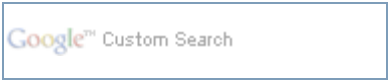
### 1. 批量差异备份

/etc/cron.daily/backup

```
#!/bin/bash
for ip in {23..32} {49,50} {81..92}; do mkdir -p /opt/backup/$ip/srv/; rsync -auz --delete
root@172.16.3.$ip:/srv/* /opt/backup/$ip/srv/; done &
for ip in {23..32} {49,50} {81..92}; do mkdir -p /opt/backup/$ip/www/; rsync -auz --delete
root@172.16.3.$ip:/www/* /opt/backup/$ip/www/; done &
```



2. tar 备份



# 第 12 章 信息安全

## 目录

[1. 用户注册与登录安全](#)

[2. 目录文件与权限](#)

[2.1. 读写权限](#)

[2.2. 访问权限](#)

[3. 密码安全](#)

[4. 注入检查](#)

[5. 防止刷新与重复POST](#)

[6. 屏蔽出错信息](#)

[6.1. 屏蔽php出错信息](#)

[7. 环境安全](#)

[7.1. 开发环境](#)

[7.2. 测试环境](#)

[7.3. 程序](#)

[7.4. 数据库](#)

SQL注入，OS命令注入，缓冲溢出、跨站脚本、缺少验证、缺少认证、使用硬编码证书、敏感数据忘记加密、不受限制上传文件类型、依赖不可信的输入、用不必要的高级权限执行任务、跨站请求伪造....

## 1. 用户注册与登录安全

用户注册与登录除了使用图片验证外，还应该记录来源IP，同时限制用户使用自动注册工具



3. 密码安全

虽然md5摘要算法作为密码仍不能保证安全。我一般采用加入干扰词的方法避免被猜中

```
password = md5/sha1(password + salt)
```