Netkiller Linux Web 手札

Apache, Lighttpd, Nginx, Resin, Tomcat, Jboss, Zope...

Mr. Neo Chan, 陈景峰

中国广东省深圳市宝安区龙华镇

518109

+86 755 29812080

+86 755 29812080
<openunix@163.com>

文档出处: http://netkiller.sourceforge.net/ | http://netkiller.github.com

文档最近一次更新于 Fri Nov 18 02:07:04 UTC 2011

2010-11-18

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:

Netkiller Architect 手札	Netkiller Linux 手札	Netkiller Developer 手札	Netkiller Database 手札
Netkiller Debian 手札	Netkiller CentOS 手札	Netkiller FreeBSD 手札	Netkiller Shell 手札
Netkiller Web 手札	Netkiller Monitoring 手札	Netkiller Storage 手札	Netkiller Mail System 手札
Netkiller MySQL 手札	Netkiller LDAP 手札	Netkiller Security 手札	Netkiller Version 手札
Netkiller Intranet 手札	Netkiller Cisco IOS 手札	Netkiller Writer 手札	Netkiller Studio Linux 手札

目录

范例清单

下一页

自述

Home | Mirror | Search

Google™ Custom Search Search

自述

目录

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试的写作，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

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

陈景峰 (ㄔㄣ ㄐㄧㄥ ㄈㄥ)

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

IRC irc.freenode.net #ubuntu / #ubuntu-cn

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 上一级 下一页

自述 起始页 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

第 1 章 web 服务器排名

http://news.netcraft.com/

第 2 章 LAMP

目录

1. Install

1.1. Quick install apache with aptitude

$ sudo apt-get install apache2$ sudo apt-get install apache2-mpm-worker

```
netkiller@Linux-server:~$ sudo apt-get install apache2
```

1.1.1. command

enable module: a2enmod

enable site: a2ensite

1.1.2. rewrite module

```
$ sudo a2enmod rewrite
```

1.1.3. PHP module

```
$ sudo a2enmod php5
```

1.1.4. deflate module

```
root@neo:/etc/apache2# a2enmod deflate
Module deflate installed; run /etc/init.d/apache2 force-reload to enable.
root@neo:/etc/apache2# /etc/init.d/apache2 force-reload
 * Forcing reload of apache 2.0 web server...                              [
ok ]
root@neo:/etc/apache2#
```

1.1.5. ssl module

a2enmod ssl

a2ensite ssl

/etc/apache2/httpd.conf 加入

```
ServerName 220.201.35.11
```

安全模块

```
netkiller@Linux-server:~$ sudo apt-get install libapache2-mod-security

netkiller@Linux-server:/etc/apache2$ sudo vi ports.conf
netkiller@Linux-server:/etc/apache2$ cat ports.conf
Listen 80
Listen 443

NameVirtualHost *
NameVirtualHost *:443


netkiller@Linux-server:/etc/apache2$ sudo apache2-ssl-certificate
or
netkiller@Linux-server:~$ apache2-ssl-certificate -days 365

netkiller@Linux-server:~$ a2enmod ssl
or
netkiller@Linux-server:/etc/apache2/mods-enabled$ sudo ln -s ../mods-available/ssl.conf
netkiller@Linux-server:/etc/apache2/mods-enabled$ sudo ln -s ../mods-available/ssl.load

netkiller@Linux-server:/etc/apache2/sites-enabled$ sudo mkdir ssl/
netkiller@Linux-server:/etc/apache2/sites-enabled$ sudo cp netkiller woodart ssl/


netkiller@Linux-server:/etc/apache2/mods-enabled$ sudo /etc/init.d/apache2 reload
 * Reloading apache 2.0 configuration...                                        [ ok ]
netkiller@Linux-server:/etc/apache2/mods-enabled$
```

1.1.6. VirtualHost

VirtualHost 虚拟主机

```

netkiller@Linux-server:/etc/apache2/sites-available$ sudo vi woodart

#NameVirtualHost neo.6600.org
<VirtualHost 220.201.35.11>
        ServerAdmin openx@163.com

        DocumentRoot /home/netkiller/www
        ServerName neo.6600.org
        ServerAlias www.neo.6600.org
        <Directory /home/netkiller/www>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride All
                Order allow,deny
                allow from all
                # Uncomment this directive is you want to see apache2's
                # default start page (in /apache2-default) when you go to /
                #RedirectMatch ^/$ /apache2-default/
        </Directory>

#       ScriptAlias /cgi-bin/ /home/netkiller/www/
#       <Directory "/home/netkiller/www">
#               AllowOverride None
#               Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
#               Order allow,deny
#               Allow from all
#       </Directory>

        ErrorLog /var/log/apache2/neo.error.log

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
#       LogLevel warn

        CustomLog /var/log/apache2/neo.access.log combined
#       ServerSignature On
```

```
</VirtualHost>

netkiller@Linux-server:/etc/apache2/sites-available$ sudo apache2 -k restart
```

1.1.7. ~userdir module - /public_html

~web环境

```
netkiller@Linux-server:~$ mkdir public_html
netkiller@Linux-server:~$ cd public_html/
netkiller@Linux-server:~/public_html$
netkiller@Linux-server:~/public_html$ echo helloworld>index.html
netkiller@Linux-server:~/public_html$ ls
index.html
```

http://xxx.xxx.xxx.xxx/~netkiller/

1.2. PHP 5

$ sudo apt-get install php5

```
netkiller@Linux-server:~$ sudo apt-get install php5
```

pgsql模块

```
netkiller@Linux-server:~$ sudo apt-get install php5-pgsql

netkiller@Linux-server:~$sudo cp /usr/lib/php5/20051025/pgsql.so /etc/php5/apache2/
```

php5-gd - GD module for php5

$ sudo apt-get install php5-gd

```
netkiller@Linux-server:~$ apt-cache search gd
libgdbm3 - GNU dbm database routines (runtime version)
libgd2-xpm - GD Graphics Library version 2
php5-gd - GD module for php5
pnm2ppa - PPM to PPA converter
postgresql-doc-8.1 - documentation for the PostgreSQL database management system
libruby1.8 - Libraries necessary to run Ruby 1.8
ruby1.8 - Interpreter of object-oriented scripting language Ruby 1.8
klogd - Kernel Logging Daemon
sysklogd - System Logging Daemon
upstart-logd - boot logging daemon
netkiller@Linux-server:~$ sudo apt-get install php5-gd

netkiller@Linux-server:~$
```

1.3. Compile and then install Apache

1.3.1. Apache 安装与配置

configure

--with-mpm=worker 进程,线程混合方式效率提高不少

--enable-modules='dir mime' 没有它就找不到index.*文件

--enable-rewrite=shared Rewrite用于表态化

--enable-expires=shared 禁止页面被 cache

--enable-authz_host=shared Order权限

--enable-setenvif=shared

--enable-log_config=shared 日志格式

--enable-speling=shared 允许自动修正拼错的URL

--enable-deflate=shared 压缩传送

--enable-mods-shared='cache file-cache disk-cache mem-cache proxy proxy-ajp proxy-balancer' 代理和缓存

```
tar zxvf httpd-2.2.4.tar.gz
cd httpd-2.2.4
./configure --prefix=/usr/local/httpd-2.2.4 \
--with-mpm=worker \
--enable-modules='dir mime' \
--enable-rewrite=shared \
--enable-authz_host=shared \
--enable-alias=shared \
--enable-setenvif=shared \
--enable-log_config=shared \
--enable-speling=shared \
--enable-filter=shared \
--enable-deflate=shared \
--enable-headers=shared \
--enable-expires=shared \
--enable-mods-shared='cache file-cache disk-cache mem-cache proxy proxy-ajp proxy-balancer' \
--disable-include \
--disable-actions \
--disable-alias \
--disable-asis \
--disable-autoindex \
--disable-auth_basic \
--disable-authn_file \
--disable-authn_default \
--disable-authz_groupfile \
--disable-authz_user \
--disable-authz_default \
--disable-cgi \
--disable-cgid \
--disable-env \
--disable-negotiation \
--disable-status \
--disable-userdir
```

make; make install

启动

```
ln -s /usr/local/httpd-2.2.4/ /usr/local/apache

/usr/local/httpd/bin/apachectl start
```

1.3.2. 优化编译条件

```
# vim server/mpm/worker/worker.c

# define DEFAULT_SERVER_LIMIT 256
# define MAX_SERVER_LIMIT 20000
# define DEFAULT_THREAD_LIMIT 512
# define MAX_THREAD_LIMIT 20000
```

1.3.3. PHP

过程 2.1. 安装PHP

1. 第一步

```
cd /usr/local/src
wget http://cn2.php.net/get/php-5.3.0.tar.bz2/from/cn.php.net/mirror
tar jxvf php-5.3.0.tar.bz2
cd php-5.3.0
```

2. 第二步

```
./configure --prefix=/usr/local/php-5.3.0 \
--with-config-file-path=/usr/local/php-5.3.0/etc \
--with-apxs2=/usr/local/apache/bin/apxs \
--with-curl \
--with-gd \
--with-ldap \
--with-snmp \
--enable-zip \
--enable-exif \
--with-libxml-dir \
--with-mysql \
--with-mysqli \
--with-pdo-mysql \
--with-pdo-pgsql

make
make test
make install
```

   a. 建立符号连接

   ```
   ln -s /usr/local/php-5.3.0 /usr/local/php
   ```

   b. php.ini

   ```
   cp php.ini-dist /usr/local/php/etc/php.ini
   ```

   c. conf/httpd.conf

   ```
   AddType application/x-httpd-php .php .phtml
   AddType application/x-httpd-php-source .phps
   ```

   reload apache

3. 最后一步

phpinfo() 测试文件复杂到apache目录

例 2.1. index.php

```
<?php phpinfo(); ?>
```

--with-snmp

redhat as4 启用 --with-snmp 需要安装下面包

```
rpm -i elfutils-libelf-devel-0.97.1-3.i386.rpm
rpm -i elfutils-devel-0.97.1-3.i386.rpm
```

```
rpm -i beecrypt-devel-3.1.0-6.i386.rpm
rpm -i net-snmp-devel-5.1.2-11.EL4.7.i386.rpm
```

1.3.4. Automation Installing

例 2.2. autolamp.sh

```
#!/bin/bash
HTTPD_SRC=httpd-2.2.15.tar.gz
PHP_SRC=php-5.2.13.tar.gz
MYSQL_SRC='mysql-5.1.45.tar.gz'
MYSQL_LIBS_SRC='mysql-5.1.45-linux-x86_64-glibc23.tar.gz'

SRC_DIR=$(pwd)
HTTPD_DIR=${HTTPD_SRC%%.tar.gz}
PHP_DIR=${PHP_SRC%%.tar.*}
MYSQL_DIR=${MYSQL_SRC%%.tar.*}
MYSQL_LIBS_DIR=${MYSQL_LIBS_SRC%%.tar.*}

function clean(){
        rm -rf $HTTPD_DIR
        rm -rf $PHP_DIR
        rm -rf $MYSQL_DIR
        rm -rf $MYSQL_LIBS_DIR
}
function mysql(){
rm -rf $MYSQL_DIR
tar zxf $MYSQL_SRC
cd $MYSQL_DIR
./configure \
--prefix=/usr/local/$MYSQL_DIR \
--with-mysqld-user=mysql \
--with-unix-socket-path=/tmp/mysql.sock \
--with-charset=utf8 \
--with-collation=utf8_general_ci \
--with-pthread \
--with-mysqld-ldflags \
--with-client-ldflags \
--with-openssl \
--without-docs \
--without-debug \
--without-ndb-debug \
--without-bench
#-–without-isam
#--without-innodb \
#--without-ndbcluster \
#--without-blackhole \
#--without-ibmdb2i \
#--without-federated \
#--without-example \
#--without-comment \
#--with-extra-charsets=gbk,gb2312,utf8 \

#--localstatedir=/usr/local/mysql/data
#--with-extra-charsets=all
make clean
make && make install
cd ..
/usr/local/$MYSQL_DIR/bin/mysql_install_db
}
function httpd(){
rm -rf $HTTPD_DIR
tar zxf $HTTPD_SRC
cd $HTTPD_DIR
./configure --prefix=/usr/local/$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-status \
--disable-autoindex \
--disable-asis \
--disable-cgi \
--disable-cgid \
--disable-negotiation \
--disable-actions \
--disable-userdir \
--disable-alias

make clean
make && make install
cd ..
}
function php(){
rm -rf $MYSQL_LIBS_DIR
tar zxf $MYSQL_LIBS_SRC
rm -rf $PHP_DIR
tar zxf $PHP_SRC
cd $PHP_DIR

./configure --prefix=/usr/local/$PHP_DIR \
--with-config-file-path=/usr/local/$PHP_DIR/etc \
--with-apxs2=/usr/local/$HTTPD_DIR/bin/apxs \
--with-curl \
--with-gd \
--with-jpeg-dir=/usr/lib64 \
--with-iconv \
--with-zlib-dir \
--with-pear \
--with-libxml \
--with-dom \
--with-xmlrpc \
--with-openssl \
--with-mysql=/usr/local/mysql-5.1.45-linux-x86_64-glibc23 \
--with-mysqli \
--with-pdo-mysql \
--enable-memcache \
--enable-zip \
--enable-sockets \
--enable-soap \
--enable-mbstring \
--enable-magic-quotes \
--enable-inline-optimization \
--enable-xml

#make && make test && make install
make &&  make install
cp /usr/local/src/$PHP_DIR/php.ini-dist /usr/local/$PHP_DIR/php.ini
}
function depend(){
        yum install gcc gcc-c++ -y
        yum install -y libxml2-devel libxslt-devel
        yum install curl-devel -y
        yum install gd-devel libjpeg-devel libpng-devel -y
        yum install ncurses-devel -y
        yum install mysql-devel -y
        yum install libevent-devel -y
}
function java(){
        #yum install java-1.6.0-openjdk -y
        chmod +x jdk-6u20-linux-x64.bin
        ./jdk-6u20-linux-x64.bin
        mv jdk1.6.0_20 ..
        ln -s /usr/local/jdk1.6.0_20 /usr/local/java
}
function memcached(){
        MEMCACHED_PKG=memcached-1.4.5.tar.gz
        MEMCACHED_SRC=memcached-1.4.5
        rm -rf $MEMCACHED_SRC
        tar zxf $MEMCACHED_PKG
        cd $MEMCACHED_SRC
        ./configure --prefix=/usr/local/memcached-1.4.5
        make && make install
}
# 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
        ;;
  memcached)
        memcached
        ;;
  all)
        clean

        echo ##################################################
        echo # $MYSQL_DIR Installing...
        echo ##################################################
        mysql

        echo ##################################################
        echo # $HTTPD_DIR Installing...
        echo ##################################################
        httpd

        echo ##################################################
        echo # $PHP_DIR Installing...
        echo ##################################################
        php

        ln -s /usr/local/$HTTPD_DIR /usr/local/apache
        ln -s /usr/local/$MYSQL_DIR /usr/local/mysql
        ln -s /usr/local/$PHP_DIR /usr/local/php

        clean
        ;;
  *)
        echo $"Usage: $0 {httpd|php|mysql|all|clean}"
        RETVAL=2
        ;;
esac

exit $RETVAL
```

1.4. XAMPP

1.4.1. XAMPP for Linux

http://www.apachefriends.org/en/xampp-linux.html

install

```
tar xvfz xampp-linux-1.7.3a.tar.gz -C /opt
```

start

```
/opt/lampp/lampp start
```

stop

```
/opt/lampp/lampp stop
```

remove

```
rm -rf /opt/lampp
```

1.4.2. php5

```
./lampp php5
XAMPP: PHP 5.3.8 already active.

./lampp startapache
XAMPP: Starting Apache with SSL (and PHP5)...

./lampp startmysql
XAMPP: Starting MySQL...
```

上一页 下一页

第 1 章 web 服务器排名 起始页 2. Module

2. Module

```
模块的做用如下:
mod_access        提供基于主机的访问控制命令
mod_actions 能够运行基于MIME类型的CGI脚本或HTTP请求方法
mod_alias         能执行URL重定向服务
mod_asis          使文档能在没有HTTP头标的情况下被发送到客户端
mod_auth          支持使用存储在文本文件中的用户名、口令实现认证
mod_auth_dbm 支持使用DBM文件存储基本HTTP认证
mod_auth_mysql 支持使用MySQL数据库实现基本HTTP认证
mod_auth_anon 允许以匿名方式访问需要认证的区域
mod_auth_external支持使用第三方认证
mod_autoindex 当缺少索引文件时，自动生成动态目录列表
mod_cern_meta 提供对元信息的支持
mod_cgi           支持CGI
mod_dir           能够重定向任何对不包括尾部斜杠字符命令的请求
mod_env           使你能够将环境变量传递给CGI或SSI脚本
mod_expires 让你确定Apache在服务器响应请求时如何处理Expires
mod_headers 能够操作HTTP应答头标
mod_imap          提供图形映射支持
mod_include 使支持SSI
mod_info          对服务器配置提供了全面的描述
mod_log_agent 允许在单独的日志文件中存储用户代理的信息
mod_log_config 支持记录日志
mod_log_referer 提供了将请求中的Referer头标写入日志的功能
mod_mime 用来向客户端提供有关文档的元信息
mod_negotiation 提供了对内容协商的支持
mod_setenvif 使你能够创建定制环境变量
mod_speling 使你能够处理含有拼写错误或大小写错误的URL请求
mod_status        允许管理员通过WEB管理Apache
mod_unique_id 为每个请求提供在非常特殊的条件下保证是唯一的标识
```

常用模块

```
LoadModule dir_module          modules/mod_dir.so
LoadModule mime_module         modules/mod_mime.so
LoadModule expires_module      modules/mod_expires.so
LoadModule config_log_module   modules/mod_log_config.so
LoadModule alias_module        modules/mod_alias.so
LoadModule rewrite_module      modules/mod_rewrite.so
LoadModule access_module       modules/mod_access.so
LoadModule auth_module         modules/mod_auth.so
```

2.1. Output a list of modules compiled into the server.

This will not list dynamically loaded modules included using the LoadModule directive.

```
[root@development bin]# httpd -l
Compiled in modules:
  core.c
  worker.c
  http_core.c
  mod_so.c
```

2.2. Core

2.2.1. Listen

绑定多个IP

```
#Listen 80
Listen 192.168.3.40:80
Listen 192.168.4.40:80
Listen 192.168.5.40:80
```

2.2.2. Filesystem and Webspace

ref: http://httpd.apache.org/docs/2.2/en/sections.html

Filesystem Containers

```
<Directory /var/web/dir1>
        Options +Indexes
</Directory>

<Files private.html>
        Order allow,deny
        Deny from all
</Files>

<Directory /var/web/dir1>
        <Files private.html>
                Order allow,deny
                Deny from all
        </Files>
</Directory>
```

Webspace Containers

```
<LocationMatch ^/private>
        Order Allow,Deny
        Deny from all
</LocationMatch>
```

Wildcards and Regular Expressions

```
A non-regex wildcard section that changes the configuration of all user directories could look
as follows:

<Directory /home/*/public_html>
Options Indexes
</Directory>
Using regex sections, we can deny access to many types of image files at once:

<FilesMatch \.(?i:gif|jpe?g|png)$>
Order allow,deny
Deny from all
</FilesMatch>
```

2.2.2.1. Options

```
<DirectoryMatch (/var/www/logs|/var/www/logs/*)>
        Options FollowSymLinks MultiViews Indexes

        DirectoryIndex index.html

        AllowOverride AuthConfig
        Order Allow,Deny
        Allow From All

        AuthName "Logs Access"
        AuthType Basic
        AuthUserFile /etc/nagios3/htpasswd.users
        require valid-user
</DirectoryMatch>
```

1. None是禁止所有

2. Indexes 当没有index.html 的时候列出目录

3. FollowSymLinks 允许符号连接，可以通过符号连接跨越DocumentRoot

4. AllowOverride 定义是否允许各个目录用目录中的.htaccess覆盖这里设定的Options

5.

2.2.3. Etag

```
<Directory /www>
        <Files ~ "\.(gif|jpe?g|png|html|css|js)$">
                FileETag INode MTime Size
        </Files>
</Directory>
```

2.2.4. 隐藏 Apache 版本信息

```
ServerTokens ProductOnly
ServerSignature Off
```

2.3. worker

worker

```
# Server-pool management (MPM specific)
Include conf/extra/httpd-mpm.conf
```

conf/extra/httpd-mpm.conf

mpm_worker_module

```
<IfModule mpm_worker_module>
    ServerLimit          64
    ThreadLimit          256
    StartServers         8
    MaxClients           15000
    MinSpareThreads      100
    MaxSpareThreads      200
    ThreadsPerChild      256
    MaxRequestsPerChild   10000
</IfModule>
```

```
ServerLimit 默认是16，它决定系统最多启动几个httpd进程。
ThreadLimit 默认是64，
ThreadsPerChild* ServerLimit=系统支持的最大并发。
MaxClients<ThreadsPerChild* ServerLimit，MaxClients如果大于400将被限制在400.
400只是理论最大并发，实际并发就是MaxClients的值。
理论并发有什么用我不知道。


指令说明：
        StartServers：设置服务器启动时建立的子进程数量。因为子进程数量动态的取决于负载的轻重,所有一般没有必要调
整这个参数。
        ServerLimit：服务器允许配置的进程数上限。只有在你需要将MaxClients和ThreadsPerChild设置成需要超过默
认值16个子进程的时候才需要使用这个指令。不要将该指令的值设置的比MaxClients 和ThreadsPerChild需要的子进程数量
高。修改此指令的值必须完全停止服务后再启动才能生效，以restart方式重启动将不会生效。
        ThreadLimit：设置每个子进程可配置的线程数ThreadsPerChild上限,该指令的值应当和ThreadsPerChild可能达
到的最大值保持一致。修改此指令的值必须完全停止服务后再启动才能生效，以restart方式重启动将不会生效。
        MaxClients：用于伺服客户端请求的最大接入请求数量（最大线程数）。任何超过MaxClients限制的请求都将进入等
候队列。默认值是"400"，16 (ServerLimit)乘以25(ThreadsPerChild)的结果。因此要增加MaxClients的时候，你必须同
时增加 ServerLimit的值。笔者建议将初始值设为(以Mb为单位的最大物理内存/2),然后根据负载情况进行动态调整。比如一
台4G内存的机器，那么初始值就是4000/2=2000。
        MinSpareThreads：最小空闲线程数,默认值是"75"。这个MPM将基于整个服务器监视空闲线程数。如果服务器中总的
空闲线程数太少，子进程将产生新的空闲线程。
        MaxSpareThreads：设置最大空闲线程数。默认值是"250"。这个MPM将基于整个服务器监视空闲线程数。如果服务器
中总的空闲线程数太多，子进程将杀死多余的空闲线程。MaxSpareThreads的取值范围是有限制的。Apache将按照如下限制自动
修正你设置的值：worker要求其大于等于 MinSpareThreads加上ThreadsPerChild的和。
        ThreadsPerChild：每个子进程建立的线程数。默认值是25。子进程在启动时建立这些线程后就不再建立新的线程了。
```

```
每个子进程所拥有的所有线程的总数要足够大，以便可以处理可能的请求高峰。
        MaxRequestsPerChild：设置每个子进程在其生存期内允许伺服的最大请求数量。到达MaxRequestsPerChild的限
制后，子进程将会结束。如果MaxRequestsPerChild为"0"，子进程将永远不会结束。将MaxRequestsPerChild设置成非零值
有两个好处：可以防止(偶然的)内存泄漏无限进行而耗尽内存；
给进程一个有限寿命，从而有助于当服务器负载减轻的时候减少活动进程的数量。
如果设置为非零值，笔者建议设为10000-30000之间的一个值。
         公式：
         ThreadLimit >= ThreadsPerChild
         MaxClients <= ServerLimit * ThreadsPerChild 必须是ThreadsPerChild的倍数
         MaxSpareThreads >= MinSpareThreads+ThreadsPerChild
```

2.4. Apache Log

2.4.1. LogLevel

日志级别

语法：LogLevel level

```
可以选择下列level，依照重要性降序排列：
emerg  紧急(系统无法使用)
alert  必须立即采取措施
crit    致命情况
error  错误情况
warn    警告情况
notice  一般重要情况
info   普通信息
debug  调试信息
```

```
LogLevel crit
```

2.4.2. LogFormat

分割log日志文件

```

<IfModule log_config_module>
    #
    # The following directives define some format nicknames for use with
    # a CustomLog directive (see below).
    #
    #LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %{email}C
%{nickname}C" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common

    <IfModule logio_module>
      # You need to enable mod_logio.c to use %I and %O
      LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O" combinedio
    </IfModule>

    #
    # The location and format of the access logfile (Common Logfile Format).
    # If you do not define any access logfiles within a <VirtualHost>
    # container, they will be logged here.  Contrariwise, if you *do*
    # define per-<VirtualHost> access logfiles, transactions will be
    # logged therein and *not* in this file.
    #
    #CustomLog logs/access_log common

    #
    # If you prefer a logfile with access, agent, and referer information
    # (Combined Logfile Format) you can use the following directive.
    #
    CustomLog logs/access_log combined

    #CookieLog logs/cookie_log
</IfModule>
```

2.4.3. Compressed

```

# compressed logs
$ CustomLog "|/usr/bin/gzip -c >> /var/log/access_log.gz" common
```

2.4.4. rotatelogs - Piped logging program to rotate Apache logs

rotatelogs是一个配合Apache管道日志功能使用的简单程序。举例:

```
rotatelogs logfile [ rotationtime [ offset ]] | [ filesizeM ]

选项
logfile
它加上基准名就是日志文件名。如果logfile中包含'%'，则它会被视为用于的strftime(3)的格式字串；否则，它会被自动加上以秒为单位的.nnnnnnnnnn后缀。这两种格式都表示新的日志开始使用的时间。
rotationtime
日志文件回卷的以秒为单位的间隔时间
offset
相对于UTC的时差的分钟数。如果省略，则假定为0，并使用UTC时间。比如，要指定UTC时差为-5小时的地区的当地时间，则此参数应为-300。
filesizeM
指定回卷时以兆字节为单位的后缀字母M的文件大小，而不是指定回卷时间或时差。

下列日志文件格式字串可以为所有的strftime(3)实现所支持，见各种扩展库对应的strftime(3)的手册。
%A 星期名全称(本地的)
%a 3个字符的星期名(本地的)
%B 月份名的全称(本地的)
%b 3个字符的月份名(本地的)
%c 日期和时间(本地的)
%d 2位数的一个月中的日期数
%H 2位数的小时数(24小时制)
%I 2位数的小时数(12小时制)
%j 3位数的一年中的日期数
%M 2位数的分钟数
%m 2位数的月份数
%p am/pm 12小时制的上下午(本地的)
%S 2位数的秒数
%U 2位数的一年中的星期数(星期天为一周的第一天)
%W 2位数的一年中的星期数(星期一为一周的第一天)
%w 1位数的星期几(星期天为一周的第一天)
%X 时间 (本地的)
%x 日期 (本地的)
%Y 4位数的年份

CustomLog "|bin/rotatelogs /var/logs/logfile 86400" common
此配置会建立文件"/var/logs/logfile.nnnn"，其中的nnnn是名义上的日志启动时的系统时间(此时间总是滚动时间的倍数，可以用于cron脚本的同步)。在滚动时间到达时(在此例中是24小时以后)，会产生一个新的日志。

CustomLog "|bin/rotatelogs /var/logs/logfile 5M" common
此配置会在日志文件大小增长到5兆字节时滚动该日志。

ErrorLog "|bin/rotatelogs /var/logs/errorlog.%Y-%m-%d-%H_%M_%S 5M"
此配置会在错误日志大小增长到5兆字节时滚动该日志，日志文件名后缀会按照如下格式创建：errorlog.YYYY-mm-dd-HH_MM_SS

ErrorLog "| /usr/local/apache/bin/rotatelogs /www/logs/www.example.com/error_%Y_%m_%d_log 86400 480"
CustomLog "| /usr/local/apache/bin/rotatelogs /www/logs/www.example.com/access_%Y_%m_%d_log 86400 480" common

CustomLog "|/usr/local/httpd/bin/rotatelogs /www/logs/www.example.com/access.%Y-%m-%d.log 86400 480" combined
```

2.4.5. cronolog

cronolog

```
cd /usr/local/src/
wget http://cronolog.org/download/cronolog-1.6.2.tar.gz
tar zxvf cronolog-1.6.2.tar.gz
cd cronolog-1.6.2
./configure --prefix=/usr/local/cronolog
make
make install
```

CustomLog "|/usr/local/cronolog/sbin/cronolog /opt/apache/logs/access_log.%Y%m%d" combined

2.4.6. 日志合并

合并多个服务器的日志文件（如log1、log2、log3），并输出到log_all中的方法是：

```
 $ sort -m -t " " -k 4 -o log_all log1 log2 log3
```

2.4.7. 日志归档

```
30 4 * * * /usr/bin/gzip -f /www/logs/access.`date -d yesterday +%Y-%m-%d`.log
```

2.4.8. logger

https://www.sit.auckland.ac.nz/Logging_to_syslog_with_Apache

```

Logging to syslog with Apache

First you will need to install syslog-ng. This is the logging server that will send the log
data to the syslog box.

apt-get update && apt-get install syslog-ng
syslog-ng uses a socket device to accept data from apache or whatever program is creating the
logs.

Use the configuration here: Syslog-ng default config.

The first part indicates what the socket will be called and where it will live. The second part
tells syslog-ng where to send the collected data. The restart syslog-ng (/etc/init.d/syslog-ng
restart)l.

Configure apache's logging

Add these directives to send apache's logs via a socket to syslog

CustomLog "|/usr/bin/logger -s -t 'monitor.cs.auckland.ac.nz' -p info -u
/var/log/apache_log.socket" Combined
ErrorLog "|/usr/bin/logger -s -t 'monitor.cs.auckland.ac.nz' -p err -u
/var/log/apache_log.socket"
Apache will then use the logger program to send data to syslog. /var/log/apache_log.socket
refers to the device that syslog-ng has created. Data sent to this device is sent over the
network to the main syslog box.

Troubleshooting

It seems that apache 2.0.54-5 does not like logging to a file and to a process at the same
time. In this case log entries will become re-ordered or missed out. You can use the test
scripts below to check if this is happening.

Testing

Here are some useful scripts that can help with testing to make sure the logging is working as
expected.

You can simulate http accesses using lynx with this command:

watch lynx -source http://monitor.cs.auckland.ac.nz/
Which will make a http request every two seconds. Or, for a better test:

for i in `seq 1 100`; do lynx -source http://monitor.cs.auckland.ac.nz/$i;sleep 3;done
The result of this test is a sequence of log entires from 1 to 100. If entries are missing or
in the wrong order, you know there is a problem.

```

2.4.9. other

```
CustomLog "|/usr/bin/your_script" Combined
ErrorLog "|/usr/bin/your_script"
```

2.5. mod_access

```

<Directory /www>
  Order Allow,Deny
</Directory>

<Directory /www>
        Order Deny,Allow
        Deny from all
        Allow from apache.org
</Directory>


<Directory /www>
        Order Allow,Deny
        Allow from apache.org
        Deny from foo.apache.org
</Directory>

```

```
A (partial) domain-name
Example: Allow from apache.org

A full IP address
Example: Allow from 10.1.2.3

A partial IP address
Example: Allow from 10.1

A network/netmask pair
Example: Allow from 10.1.0.0/255.255.0.0

A network/nnn CIDR specification
Example: Allow from 10.1.0.0/16
```

```
<DirectoryMatch (/usr/share/nagios3/htdocs|/usr/lib/cgi-bin/nagios3|/etc/nagios3/stylesheets)>
        Options FollowSymLinks

        DirectoryIndex index.html

        AllowOverride AuthConfig
        Order Allow,Deny
        Allow From All

        AuthName "Nagios Access"
        AuthType Basic
        AuthUserFile /etc/nagios3/htpasswd.users
        # nagios 1.x:
        #AuthUserFile /etc/nagios/htpasswd.users
        require valid-user
</DirectoryMatch>
```

2.6. VirtualHost

conf/extra/httpd-vhosts.conf

or

/etc/httpd/conf.d/vhost.conf

```
NameVirtualHost *:80

<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot "/usr/local/httpd-2.2.14/docs/dummy-host.example.com"
    ServerName dummy-host.example.com
    ServerAlias www.dummy-host.example.com
    ErrorLog "logs/dummy-host.example.com-error_log"
    CustomLog "logs/dummy-host.example.com-access_log" common
</VirtualHost>
```

2.6.1. ServerName/ServerAlias

```
ServerName dummy-host.example.com
ServerAlias www.dummy-host.example.com
```

2.6.2. rotatelogs

```
CustomLog "|/usr/local/httpd/bin/rotatelogs /www/logs/men.xiu.com/access.%Y-%m-%d.log 86400 480" combined
ErrorLog "|/usr/local/httpd/bin/rotatelogs /www/logs/men.xiu.com/error.%Y-%m-%d.log 86400 480"
```

2.7. Alias / AliasMatch

```
Alias /image /ftp/pub/image
AliasMatch ^/icons(.*) /usr/local/apache/icons$1
```

```
cat /etc/httpd/conf.d/logs.conf

Alias /logs "/www/logs"

<Directory "/www/logs">
   Options FollowSymLinks MultiViews Indexes
   AllowOverride None
   Order allow,deny
   Allow from all
#  Order deny,allow
#  Deny from all
#  Allow from 127.0.0.1
#   AuthName "Logs Access"
#   AuthType Basic
#   AuthUserFile /etc/httpd/htpasswd.users
#   Require valid-user
</Directory>
```

2.8. Redirect / RedirectMatch

Redirect

```
Redirect /service http://foo2.example.com/service
Redirect permanent /one http://example.com/two
Redirect 303 /three http://example.com/other
```

RedirectMatch

```
RedirectMatch (.*)\.gif$ http://www.domain.com$1.jpg
```

```
<VirtualHost *:80>
     ServerName www.old.com
     DocumentRoot /path/to/htdocs
     ......
     <Directory "/path/to/htdocs">
         RedirectMatch ^/(.*)$ http://www.new.com/$1
     </Directory>
</VirtualHost>
```

2.9. Rewrite

Rewrite 需要 AllowOverride All

```
<Directory "/www">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important.  Please see
    # http://httpd.apache.org/docs/2.2/mod/core.html#options
    # for more information.
    #
    Options Indexes FollowSymLinks

    #
    # AllowOverride controls what directives may be placed in .htaccess files.
    # It can be "All", "None", or any combination of the keywords:
    #   Options FileInfo AuthConfig Limit
    #
    #AllowOverride None
    AllowOverride All

    #
    # Controls who can get stuff from this server.
    #
    Order allow,deny
    Allow from all
```

```
</Directory>
```

2.9.1. R=301

```
RewriteEngine on
RewriteCond %{HTTP_HOST} ^x.x.x.x [NC]
RewriteRule ^/(.*)$ http://www.example.com/$1 [L,R=301]
```

例 2.3. R=301

```
<VirtualHost *:80>
        ServerAdmin webmaster@example.com
        ServerName www.example.com
        ServerAlias www.second.com

    RewriteEngine On
    RewriteCond %{HTTP_HOST} ^www.example.com [NC]
    RewriteRule ^/(.*)$ http://www.other.com/$1 [L,R=301]
    RewriteCond %{HTTP_HOST} ^www.second.com [NC]
    RewriteRule ^/(.*)$ http://www.other.com/$1 [L,R=301]
</VirtualHost>
```

2.9.2. Rewrite + JkMount

JkMount 与 Rewrite 同时使用时

RewriteRule ^/communtiy/top/(.*)$ /community.do?method=activeContent&id=$1 [PT]

后面用[PT]

2.9.3. Apache redirect domain.com to www.domain.com

```
$ vi .htaccess
RewriteEngine on
RewriteCond %{HTTP_HOST} ^domain\.com
RewriteRule ^(.*)$ http://www.domain.com/$1 [R=permanent,L]
```

2.9.4. 正则匹配扩展名

```
<VirtualHost *:80>
    ServerAdmin webmaster@example.com
    DocumentRoot "/www/www.example.com/images"
    ServerName images.example.com
    RewriteEngine On
    RewriteRule ^(.+)(jpg|gif|bmp|jpeg|ico|png|css)$ http://images.other.com/$1$2 [R]
    ErrorLog "logs/images.example.com-error.log"
</VirtualHost>
```

```
<VirtualHost *:80>
        ServerAdmin webmaster@example.com
        ServerName images.example.com
        RewriteEngine On
        RewriteCond %{HTTP_HOST} ^images.example.com [NC]
        RewriteRule ^/(.*) http://images.other.com/$1 [L]
        CustomLog "|/usr/local/httpd/bin/rotatelogs /www/logs/images/access.%Y-%m-%d.log 100M"
common
</VirtualHost>
```

2.10. Proxy

```
ProxyRequests Off

<Proxy *>
        Order deny,allow
        Allow from all
</Proxy>
ProxyPass / http://your.domain.com:8080/
ProxyPassReverse / http://your.domain.com:8080/
```

2.10.1. Reverse proxy

/etc/httpd/conf.d/rails.conf

```
Listen 8080
ProxyRequests Off
<Proxy balancer://cluster>
        BalancerMember http://127.0.0.1:3001
        BalancerMember http://127.0.0.1:3002
        BalancerMember http://127.0.0.1:3003
        BalancerMember http://127.0.0.1:3004
        BalancerMember http://127.0.0.1:3005
</Proxy>

<VirtualHost *:8080>
        ServerName www.example.com:8080
        DocumentRoot /var/www/project/public
        ProxyPass /images !
        ProxyPass /stylesheets !
        ProxyPass /javascripts !
        ProxyPass / balancer://cluster/
        ProxyPassReverse / balancer://cluster/
        ProxyPreserveHost on
</VirtualHost>
```

2.11. Deflate

mod_deflate

httpd.conf中中加入下列语句:

```
<IfModule mod_deflate.c>
        SetOutputFilter DEFLATE
        DeflateCompressionLevel 9
        AddOutputFilterByType DEFLATE text/html text/plain text/xml application/x-httpd-php
        AddOutputFilter DEFLATE txt css js
        SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png)$ no-gzip dont-vary
        SetEnvIfNoCase Request_URI \.(?:exe|t?gz|zip|bz2|sit|rar)$ no-gzip dont-vary
        SetEnvIfNoCase Request_URI \.pdf$ no-gzip dont-vary
        DeflateFilterNote Input input_info
        DeflateFilterNote Output output_info
        DeflateFilterNote Ratio ratio_info
        LogFormat '"%r" %{output_info}n/%{input_info}n (%{ratio_info}n%%)' deflate
        CustomLog logs/deflate_log.log deflate
</IfModule>
```

对目录/usr/local/apache/htdocs有效

```
<Directory "/usr/local/apache/htdocs">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
        SetOutputFilter DEFLATE
        DeflateCompressionLevel 9
        AddOutputFilterByType DEFLATE text/html text/plain text/xml application/x-httpd-php
        AddOutputFilter DEFLATE txt css js
        SetEnvIfNoCase Request_URI \
        \.(?:gif|jpe?g|png)$ no-gzip dont-vary
</Directory>
```

```
<Location />
        AddOutputFilterByType DEFLATE text/html text/plain text/xml text/css text/javascript
        AddOutputFilterByType DEFLATE application/javascript application/x-javascript
application/x-httpd-php
        AddOutputFilter DEFLATE txt css js
        SetOutputFilter DEFLATE
</Location>
```

Log定义

```
DeflateFilterNote Input instream   # 未压缩前
DeflateFilterNote Output outstream # 压缩后
DeflateFilterNote Ratio ratio   # 百分比
LogFormat '"%r" %{outstream}n/%{instream}n (%{ratio}n%%)' deflate # 格式定义

CustomLog logs/deflate_log.log deflate # 日志位置
CustomLog "|/usr/local/httpd/bin/rotatelogs /www/logs/deflate.%Y-%m-%d.log 86400 480" deflate #
分割日志位置
```

2.11.1. 测试 gzip,deflate 模块

telnet www.bg7nyt.cn 80

```
GET /index.html HTTP/1.0
Host: www.bg7nyt.cn
Accept-Encoding: gzip,deflate
```

你看到的是乱码,而不是HTML.

```
curl -H Accept-Encoding:gzip,defalte http://www.example.com/index.html | gunzip
```

gunzip 可以解压压缩内容

2.12. Expires

```
ExpiresActive On
ExpiresByType image/gif "access plus 1 month"
ExpiresByType image/jpeg "access plus 1 month"
ExpiresByType image/x-icon "access plus 1 month"
ExpiresByType image/png "access plus 1 month"
ExpiresByType text/html "access plus 30 minutes"
ExpiresByType text/css  "access plus 30 minutes"
ExpiresByType text/js   "access plus 30 minutes"
ExpiresByType application/x-javascript   "access plus 30 minutes"
ExpiresByType application/x-shockwave-flash     "access plus 30 minutes"

<FilesMatch “\.(ico|jpg|jpeg|png|gif|js|css|swf|html|htm|gzip)$”>
ExpiresActive on
ExpiresDefault “access plus 2 hours”
Header set Cache-Control “max-age=1800, public”
FileETag none
</FilesMatch>
```

2.13. Cache

htcacheclean -- program for cleaning the disk cache.

2.13.1. mod_disk_cache

```
<IfModule mod_cache.c>
    CacheDefaultExpire 86400
    <ifModule mod_disk_cache.c>
        CacheEnable disk /
        CacheRoot /tmp/apacheCache
        CacheDirLevels 5
        CacheDirLength 5
        CacheMaxFileSize 1048576
        CacheMinFileSize 10
    </ifModule mod_disk_cache.c>
</IfModule mod_cache.c>
```

2.13.2. mod_mem_cache

```
<IfModule mod_cache.c>
    <ifModule mod_mem_cache.c>
        CacheEnable mem /
        MCacheMaxObjectCount 20000
        MCacheMaxObjectSize 1048576
        MCacheMaxStreamingBuffer 65536
        MCacheMinObjectSize 10
        MCacheRemovalAlgorithm GDSF
        MCacheSize 131072
    </ifModule mod_disk_cache.c>
</IfModule mod_cache.c>
```

2.14. usertrack

跟踪用户信息

跟踪用户的cookie,使用log日志文件记录用户的cookie

```
LoadModule usertrack_module modules/mod_usertrack.so

CookieTracking on
CookieDomain .example.com
CookieExpires "10 years"
CookieStyle Cookie

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %{cookie}n" combined
```

2.15. Charset

Default charset

```
AddCharset UTF-8 .html

AddType 'text/html; charset=UTF-8' html

AddDefaultCharset UTF-8
```

Files match

```
<FilesMatch "\.(htm|html|css|js)$">
        ForceType 'text/html; charset=UTF-8'
</FilesMatch>

<FilesMatch "\.(htm|html|css|js)$">
        AddDefaultCharset UTF-8
</FilesMatch>
```

Changing the occasional file

```
<Files "example.html">
        AddCharset UTF-8 .html
</Files>
```

```
<Files "example.html">
        ForceType 'text/html; charset=UTF-8'
</Files>
```

2.16. Dir

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

2.17. Includes

```
<Directory "/www">
        Options Indexes FollowSymLinks +Includes
</Directory>
```

```
<IfModule mime_module>
        AddType text/html .shtml
    AddOutputFilter INCLUDES .shtml
</IfModule>
```

2.18. Apache Status

开启Apache的status模块，需要修改httpd.conf，增加以下配置段：

```
ExtendedStatus On
<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from 125.76.229.113
</Location>
```

http://www.domain.com/server-status

2.19. Mod Perl

ref: http://search.cpan.org/~agrundma/Catalyst-Engine-Apache-1.07/lib/Catalyst/Engine/Apache2/MP20.pm

$ sudo apt-get install libapache2-mod-perl2 $ sudo apt-get install libcatalyst-engine-apache-perl

```
$ sudo vi /etc/apache2/sites-available/catalyst.conf
```

例 2.4. mod_perl.conf

```
PerlSwitches -I/var/www/MyApp/lib
# Preload your entire application
PerlModule MyApp

<VirtualHost 192.168.245.129:80>
        ServerName 192.168.245.129
        DocumentRoot /var/www/MyApp/root

        <Directory /var/www/MyApp/root>
                Options Indexes FollowSymLinks
                AllowOverride None
                Order allow,deny
                Allow from all
        </Directory>

        # If the server is started as:
        #       httpd -X -D PERLDB
        # then debugging will be turned on
#       <IfDefine PERLDB>
#               PerlRequire conf/db.pl
#               <Location />
#                       PerlFixupHandler Apache::DB
#               </Location>
#       </IfDefine>

        <Location />
                SetHandler modperl
                PerlResponseHandler MyApp
        </Location>

        Alias /static /var/www/MyApp/root/static
        <Location /static>
                SetHandler default-handler
        </Location>
</VirtualHost>
```

db.pl

```
use APR::Pool ();
use Apache::DB ();
Apache::DB->init();
```

enable site

```
$ sudo a2ensite mod_perl.conf
$ sudo /etc/init.d/apache2 restart
```

2.20. Module FAQ

```
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 358 of /etc/httpd/conf/httpd.conf:
Invalid command 'Order', perhaps mis-spelled or defined by a module not included
in the server configuration
[FAILED]
LoadModule access_module /etc/httpd/modules/mod_access.so
LoadModule auth_module /etc/httpd/modules/mod_auth.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 368 of /etc/httpd/conf/httpd.conf:
Invalid command 'UserDir', perhaps mis-spelled or defined by a module not includ
ed in the server configuration
[FAILED]
LoadModule userdir_module /etc/httpd/modules/mod_userdir.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 396 of /etc/httpd/conf/httpd.conf:
Invalid command 'DirectoryIndex', perhaps mis-spelled or defined by a module not
included in the server configuration
[FAILED]
LoadModule dir_module /etc/httpd/modules/mod_dir.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 419 of /etc/httpd/conf/httpd.conf:
Invalid command 'TypesConfig', perhaps mis-spelled or defined by a module not in
cluded in the server configuration
[FAILED]
LoadModule mime_module /etc/httpd/modules/mod_mime.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 491 of /etc/httpd/conf/httpd.conf:
Invalid command 'LogFormat', perhaps mis-spelled or defined by a module not incl
uded in the server configuration
[FAILED]
LoadModule log_config_module /etc/httpd/modules/mod_log_config.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 555 of /etc/httpd/conf/httpd.conf:
Invalid command 'Alias', perhaps mis-spelled or defined by a module not included
in the server configuration
[FAILED]
LoadModule alias_module /etc/httpd/modules/mod_alias.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 582 of /etc/httpd/conf/httpd.conf:
Invalid command 'SetEnvIf', perhaps mis-spelled or defined by a module not inclu
ded in the server configuration
[FAILED]
LoadModule setenvif_module /etc/httpd/modules/mod_setenvif.so
[root@srv-2 modules]# /etc/init.d/httpd start
```

```
Starting httpd: Syntax error on line 636 of /etc/httpd/conf/httpd.conf:
Invalid command 'IndexOptions', perhaps mis-spelled or defined by a module not i
ncluded in the server configuration
[FAILED]
LoadModule autoindex_module /etc/httpd/modules/mod_autoindex.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd: Syntax error on line 784 of /etc/httpd/conf/httpd.conf:
Invalid command 'LanguagePriority', perhaps mis-spelled or defined by a module n
ot included in the server configuration
[FAILED]
LoadModule negotiation_module /etc/httpd/modules/mod_negotiation.so
[root@srv-2 modules]# /etc/init.d/httpd start
Starting httpd:                                            [  OK  ]
[root@srv-2 modules]#
```

2.21. mod_setenvif

屏蔽爬虫

```
<directory "/www/example.com">
    Order allow,deny
    Allow from all
    BrowserMatchNoCase "iaskspider" badguy
    BrowserMatchNoCase "QihooBot" badguy
    BrowserMatchNoCase "larbin" badguy
    BrowserMatchNoCase "iearthworm" badguy
    BrowserMatchNoCase "Outfoxbot" badguy
    BrowserMatchNoCase "lanshanbot" badguy
    BrowserMatchNoCase "Arthur" badguy
    BrowserMatchNoCase "InfoPath" badguy
    BrowserMatchNoCase "DigExt" badguy
    BrowserMatchNoCase "Embedded" badguy
    BrowserMatchNoCase "EmbeddedWB" badguy
    BrowserMatchNoCase "Wget" badguy
    BrowserMatchNoCase "CNCDialer" badguy
    BrowserMatchNoCase "LWP::Simple" badguy
    BrowserMatchNoCase "WPS" badguy
    deny from env=badguy
</directory>
```

屏蔽下载

```
BrowserMatch "NetAnt" badguy
BrowserMatch "GetRight" badguy
BrowserMatch "JetCar" badguy
BrowserMatch "Mass Downloader" badguy
BrowserMatch "ReGet" badguy
BrowserMatch "DLExpert" badguy
BrowserMatch "FlashGet" badguy
BrowserMatch "Offline Explorer" badguy
BrowserMatch "Teleport" badguy
...........

order deny,allow
deny from env=badguy
allow from all
```

3. 设置Apache实现防盗连

```
SetEnvIf Referer "http://news.netkiller.com/" local_referal
SetEnvIf Referer "$" local_referral

Order Deny,Allow
Deny from all
Allow from env=local_referal
```

配置httpd.conf文件

#LoadModule rewrite_module modules/mod_rewrite.so

去掉前面的"#"注释

AllowOverride None

改为

AllowOverride All

配置.htaccess文件

```
RewriteEngine on
RewriteCond % !^http://xxx.cn/.*$      [NC]
RewriteCond % !^http://xxx.cn$                  [NC]
RewriteCond % !^http://www.xxx.cn/.*$  [NC]
RewriteCond % !^http://www.xxx.cn$     [NC]
RewriteRule .*\.(jpg|jpeg|gif|png|bmp|rar|zip|exe)$ http://download.example.com/err.html [R,NC]
```

4. Error Prompt

4.1. Invalid command ’Order’, perhaps misspelled or defined by a module not included in the server configuration

没有加载 mod_authz_host 模块

```
LoadModule authz_host_module modules/mod_authz_host.so
```

4.2. Invalid command ’AuthUserFile’, perhaps misspelled or defined by a module not included in the server configuration

```
LoadModule auth_basic_module /usr/lib/apache2/modules/mod_auth_basic.so
LoadModule authz_owner_module /usr/lib/apache2/modules/mod_authz_owner.so
LoadModule authn_file_module  /usr/lib/apache2/modules/mod_authn_file.so
```

Home | Mirror | Search

Google™ Custom Search Search

第 3 章 Lighttpd

目录

1. 安装Lighttpd

1.1. quick install with aptitude

if you OS is Ubuntu/Debian

apt-get install lighttpd

```
netkiller@shenzhen:~$ sudo apt-get install lighttpd
```

the config file in /etc/lighttpd

```
netkiller@shenzhen:~/document/Docbook/Linux$ find /etc/lighttpd/
/etc/lighttpd/
/etc/lighttpd/lighttpd.conf
/etc/lighttpd/conf-enabled
/etc/lighttpd/conf-available
/etc/lighttpd/conf-available/10-userdir.conf
/etc/lighttpd/conf-available/10-fastcgi.conf
/etc/lighttpd/conf-available/10-cgi.conf
/etc/lighttpd/conf-available/README
/etc/lighttpd/conf-available/10-ssl.conf
```

```
/etc/lighttpd/conf-available/10-proxy.conf
/etc/lighttpd/conf-available/10-auth.conf
/etc/lighttpd/conf-available/10-simple-vhost.conf
/etc/lighttpd/conf-available/10-ssi.conf
```

Enabling and disabling modules could be done by provided e.g.

```
/usr/sbin/lighty-enable-mod fastcgi
/usr/sbin/lighty-disable-mod fastcgi
```

when you enabled a mod please force-reload it

```
netkiller@shenzhen:/etc/lighttpd$ sudo lighty-enable-mod fastcgi
Available modules: auth cgi fastcgi proxy simple-vhost ssi ssl userdir
Already enabled modules: userdir
Enabling fastcgi: ok
Run /etc/init.d/lighttpd force-reload to enable changes
netkiller@shenzhen:/etc/lighttpd$ sudo /etc/init.d/lighttpd force-reload
 * Stopping web server lighttpd
[ OK ]
 * Starting web server lighttpd
```

1.2. yum install

```
yum install lighttpd lighttpd-fastcgi -y
chkconfig lighttpd on
```

1.3. to compile and then install lighttpd

1. 下载相关软件

   立即下载

   ```
   $ sudo apt-get install libpcre3*

   cd /usr/local/src/
   wget http://www.lighttpd.net/download/lighttpd-1.4.15.tar.gz
   tar zxvf lighttpd-1.4.15.tar.gz
   cd lighttpd-1.4.15
   ```

2. 编译安装

   ```
   ./configure --prefix=/usr/local/lighttpd-1.4.15 \
   --with-bzip2 \
   --with-memcache
   make
   make install
   ```

3. 创建目录与配置文件

   ```
   ln -s /usr/local/lighttpd-1.4.15/ /usr/local/lighttpd
   mkdir -p /www/pages
   mkdir /www/logs
   mkdir /usr/local/lighttpd/htdocs
   mkdir /usr/local/lighttpd/logs
   mkdir /usr/local/lighttpd/etc
   cp ./doc/lighttpd.conf /usr/local/lighttpd/etc/
   cd /usr/local/lighttpd/
   ```

4. 配置lighttpd.conf

   vi etc/lighttpd.conf

   找到 server.modules

删除 mod_fastcgi 前的注释

跟据你的需求修改下面定义

server.document-root = "/usr/local/lighttpd/htdocs/"

server.errorlog = "/usr/local/lighttpd/logs/lighttpd.error.log"

accesslog.filename = "/usr/local/lighttpd/logs/access.log"

注释 $HTTP["url"]

```
#$HTTP["url"] =~ "\.pdf$" {
#  server.range-requests = "disable"
#}
```

5. 运行lighttpd

```
/usr/local/lighttpd/sbin/lighttpd -f /usr/local/lighttpd/etc/lighttpd.conf
```

测试

curl http://ip/ 因为/www/pages/下没有HTML页面所以返回:

404 - Not Found

1.3.1. shell script

lighttpd script

例 3.1. /etc/init.d/lighttpd

```

#!/bin/bash
# lighttpd init file for web server
#
# chkconfig: - 100 100
# description: Security, speed, compliance, and flexibility--all of these describe LightTPD
which is rapidly redefining efficiency of a webserver;
#                                as it is designed and optimized for high performance
environments.
# author: Neo Chen<openunix@163.com>
#
# processname: $PROG
# config:
# pidfile: /var/run/lighttpd

# source function library
. /etc/init.d/functions

PREFIX=/usr/local/lighttpd
PROG=$PREFIX/sbin/lighttpd
OPTIONS="-f /usr/local/lighttpd/etc/lighttpd.conf"
USER=daemon
RETVAL=0
prog="lighttpd"

start() {
        echo -n $"Starting $prog: "
        if [ $UID -ne 0 ]; then
                RETVAL=1
                failure
        else
                daemon --user=$USER $PROG $OPTIONS
                RETVAL=$?
                [ $RETVAL -eq 0 ] && touch /var/lock/subsys/lighttpd
        fi;
        echo
        return $RETVAL
}

stop() {
        echo -n $"Stopping $prog: "
        if [ $UID -ne 0 ]; then
                RETVAL=1
                failure
        else
                killproc $PROG
                RETVAL=$?
                [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/lighttpd
        fi;
        echo
        return $RETVAL
}

reload(){
        echo -n $"Reloading $prog: "
        killproc $PROG -HUP
        RETVAL=$?
        echo
        return $RETVAL
}

restart(){
        stop
        start
}

condrestart(){
    [ -e /var/lock/subsys/lighttpd ] && restart
    return 0
}

case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart)
        restart
        ;;
  reload)
        reload
        ;;
  condrestart)
        condrestart
        ;;
  status)
        status lighttpd
        RETVAL=$?
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart|reload}"
        RETVAL=1
esac

exit $RETVAL
```

2. /etc/lighttpd/lighttpd.conf

2.1. max-worker / max-fds

max-worker 我一般设置为与处理器数目相同。

max-fds 最大连接数

```
server.max-worker = 24
server.max-fds = 4096
```

2.2. accesslog.filename

通过cronolog切割日志

```
#### accesslog module
#accesslog.filename          = "/www/logs/lighttpd.access.log"
accesslog.filename = "| /usr/local/sbin/cronolog /www/logs/%Y/%m/%d/access.log"
```

2.3. ETags

disable etags

```
static-file.exclude-extensions = ( ".php", ".pl", ".fcgi" )
static-file.etags = "disable"
```

2.4. server.tag

隐藏服务器信息

```
server.tag = "Apache"
```

测试结果Server: Apache

```
curl -I http://172.16.0.7/
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 4692
Date: Fri, 04 Nov 2011 12:33:19 GMT
Server: Apache
```

3. Module

```
server.modules              = (
#                               "mod_rewrite",
#                               "mod_redirect",
#                               "mod_alias",
                                "mod_access",
#                               "mod_trigger_b4_dl",
#                               "mod_auth",
#                               "mod_status",
#                               "mod_setenv",
#                               "mod_fastcgi",
#                               "mod_proxy",
#                               "mod_simple_vhost",
#                               "mod_evhost",
#                               "mod_userdir",
#                               "mod_cgi",
#                               "mod_compress",
#                               "mod_ssi",
#                               "mod_usertrack",
#                               "mod_expire",
#                               "mod_secdownload",
#                               "mod_rrdtool",
                                "mod_accesslog"  )
```

3.1. simple_vhost

```
$ sudo lighty-enable-mod simple-vhost
```

simple-vhost.default-host = "www.example.com"

create your virtual host directory

```
$ mkdir -p /var/www/www.example.com/html
```

create a test file

```
$ echo helloworld!!!> /var/www/www.example.com/html/index.html
```

3.2. ssl

启用 ssl 模块

```
$ sudo lighttpd-enable-mod ssl
[sudo] password for neo:
Available modules: auth cgi fastcgi proxy rrdtool simple-vhost ssi ssl status userdir
Already enabled modules: cgi fastcgi simple-vhost
Enabling ssl: ok
Run /etc/init.d/lighttpd force-reload to enable changes
```

创建 ssl 证书

```
$ sudo openssl req -new -x509 -keyout server.pem -out server.pem -days 365 -nodes
$ sudo chmod 400 server.pem
```

3.3. redirect

```
url.redirect                = ( "^/music/(.+)" => "http://www.example.org/$1" )
```

301重定向

```
RewriteCond %{HTTP_HOST} ^example\.org$ [NC]
RewriteRule ^(.*)$ http://www.example.org/$1 [R=301,L]
```

lighttpd 实现上面 apache功能

```
$HTTP["host"] =~ "^example\.org" {
    url.redirect = (
        "^/(.*)$" => "http://www.example.org/$1"
    )
}

$HTTP["host"] =~ "^example\.com$" {
  url.redirect = ( "^/(.*)" => "http://www.example.com/$1" )
}
```

3.4. rewrite

example 1

```
url.rewrite-once = ( "^/wiki/(.*)$" => "/wiki/awki.cgi/$1" )
$HTTP["url"] =~ "^/wiki" {
  $HTTP["url"] !~ "^/wiki/awki.cgi/" {
    url.access-deny = ("")
  }
}
```

example 2

```
$HTTP["host"] =~ "^.*\.(example.org)$" {
  url.rewrite-once = ( "^/(.*)" => "/index.php/$1" )
}
```

example 3

```
$HTTP["host"] =~ "^.*\.(example.org)$" {
  url.rewrite = (
        "^/(images|stylesheet).*" => "/$0",
        "^/(.*)" => "/index.php/$1"
  )
}
```

3.4.1. Lighttpd Rewrite QSA

```
# Apache
RewriteRule ^/index\.html$ /index.php [QSA]
RewriteRule ^/team_(.*)\.html$ /team.php?id=$1 [QSA]

#lighttpd
"^/index\.html(.*)"                     => "/index.php$1",
"^/team_(\w+)\.html\?(.*)"              => "/team.php?id=$1&$2",
```

ref: http://redmine.lighttpd.net/wiki/lighttpd/MigratingFromApache

```
url.rewrite = (
        "^/index\.html(.*)"                     => "/index.php$1",
        "^/index\.html"                         => "/index.php",
        "^/team_(.*)\.html"                     => "/team.php?id=$1",
        "^/team_(\w+)\.html\?(.*)"              => "/team.php?id=$1&$2"
)
```

3.5. alias

```
$HTTP["host"] =~ "^.*\.(example.org)$" {
    alias.url = (
        "/images" =>
"/home/neo/workspace/Development/photography/application/photography/images",
        "/stylesheet" =>
"/home/neo/workspace/Development/photography/application/photography/stylesheet"
    )
}
```

3.6. auth

enable auth

```
$ sudo lighttpd-enable-mod auth
```

/etc/lighttpd/conf-enabled/05-auth.conf

```
$ sudo vim  conf-enabled/05-auth.conf

auth.backend = "plain"
auth.backend.plain.userfile = "/etc/lighttpd/.secret"

auth.require = ( "/tmp/" =>
        (
        "method" => "basic",
        "realm" => "Password protected area",
        "require" => "user=neo"
        )
)
```

create a passwd file

```
$ sudo vim .secret
neo:chen

$ sudo chmod 400 .secret
$ sudo chown www-data /etc/lighttpd/.secret
```

$ sudo /etc/init.d/lighttpd reload

3.7. compress

创建cache目录

```
mkdir -p /var/cache/lighttpd/compress
```

配置lighttpd.conf文件

找到server.modules列表,去掉"mod_compress"注释,再打开compress module的注释

```
#### compress module
compress.cache-dir         = "/var/lighttpd/cache/compress/"
compress.filetype          = ("text/plain", "text/html")
```

Compressing Dynamic Content¶

php.ini

```
zlib.output_compression = On
zlib.output_handler = On
```

最后使用telnet测试

telnet www.bg7nyt.cn 80

```
GET /index.html HTTP/1.0
Host: 10.10.100.183
Accept-Encoding: gzip,deflate
```

看到乱码输出,而非HTML,表示配置成功.

3.8. expire

<access|modification> <number> <years|months|days|hours|minutes|seconds>

```
expire.url = ( "/images/" => "access 1 hours" )
```

Example to include all sub-directories:

```
$HTTP["url"] =~ "^/images/" {
        expire.url = ( "" => "access 1 hours" )
}
```

3.9. status

```
$ sudo lighty-enable-mod status
$ sudo /etc/init.d/lighttpd force-reload
```

3.10. setenv

```
$HTTP["url"] =~ "^/(.*)" {
        setenv.add-response-header = ( "Cache-Control" => "no-store, no-cache, must-revalidate, post-check=0, pre-check=0, max-age=-1" )
}

$HTTP["url"] =~ ".swf" {
       setenv.add-response-header  = ("Pragma" =>"no-cache","Expires" => "-1")
}

$HTTP["url"] =~ ".swf" {
       setenv.add-response-header  = ("Cache-Control" =>"max-age=0")
}

$HTTP["url"] =~ ".html" {
       setenv.add-response-header  = ("Cache-Control" =>"s-maxage=3600")
}

$HTTP["url"] =~ ".css" {
        setenv.add-response-header = (
      "Content-Encoding" => "gzip"
    )
}
```

3.10.1. Automatic Decompression

```
  $HTTP["url"] =~ "(README|ChangeLog|\.txt)\.gz$" {
    setenv.add-response-header  = ( "Content-Encoding" => "gzip")
    mimetype.assign = ("" => "text/plain" )
  }
```

3.11. fastcgi

3.11.1. enable fastcgi

enable fastcgi

```
$ sudo lighty-enable-mod fastcgi
```

3.11.1.1. spawn-fcgi

```
#### fastcgi module
## read fastcgi.txt for more info
## for PHP don't forget to set cgi.fix_pathinfo = 1 in the php.ini
fastcgi.server             = ( ".php" =>
                                ( "localhost" =>
                                  (
                                    "socket" => "/tmp/php-fastcgi.socket",
                                    "bin-path" => "/usr/local/bin/php-cgi",
                                    "max-procs" => 16,
                                    "bin-environment" => (
                                       "PHP_FCGI_CHILDREN" => "128",
                                       "PHP_FCGI_MAX_REQUESTS" => "1000"
                                    ),
                                    "broken-scriptfilename" => "enable"
                                  )
                                )
                             )

fastcgi.server    = ( ".php" =>
        ((
                "bin-path" => "/usr/bin/php-cgi",
                "socket" => "/tmp/php.socket",
                "max-procs" => 2,
                "idle-timeout" => 200,
                "bin-environment" => (
                        "PHP_FCGI_CHILDREN" => "10",
                        "PHP_FCGI_MAX_REQUESTS" => "10000"
                ),
                "bin-copy-environment" => (
                        "PATH", "SHELL", "USER"
                ),
                "broken-scriptfilename" => "enable"
        ))
)
```

3.11.1.2. php-fpm

```
fastcgi.server = ( ".php" =>
  ( "localhost" =>
    (
      "host" => "127.0.0.1",
      "port" => "9000"
    )
  )
)
```

3.11.2. PHP

3.11.2.1. 编译安装PHP

1. 下载PHP

```
cd /usr/local/src/
wget http://cn2.php.net/get/php-5.2.3.tar.bz2/from/cn.php.net/mirror
tar jxvf php-5.2.3.tar.bz2
cd php-5.2.3
```

2. configure

```
./configure --prefix=/usr/local/php-5.2.3 \
--with-config-file-path=/usr/local/php-5.2.3/etc \
--enable-fastcgi \
--enable-force-cgi-redirect \
--with-curl \
--with-gd \
--with-ldap \
--with-snmp \
--enable-zip \
--enable-exif \
--with-pdo-mysql \
--with-pdo-pgsql \

make
make test
make install
```

其它有用的模块

```
--enable-pcntl
```

3. 符号连接

```
ln -s /usr/local/php-5.2.3 /usr/local/php
ln -s /usr/local/php/bin/php /usr/local/bin/php
```

4. php.ini

```
cp php.ini-dist /usr/local/php/etc/php.ini
```

5. env

```
PHP_FCGI_CHILDREN=384
```

6. 使用 php -v FastCGI 安装情况

php -v

显示(cgi-fcgi)表示正确

```
# cd /usr/local/php/
# bin/php -v
PHP 5.2.2 (cgi-fcgi) (built: May 25 2007 15:50:28)
Copyright (c) 1997-2007 The PHP Group
Zend Engine v2.2.0, Copyright (c) 1998-2007 Zend Technologies
```

(cgi-fcgi)不能正常工作

```
PHP 5.2.2 (cli) (built: May 25 2007 15:50:28)
Copyright (c) 1997-2007 The PHP Group
Zend Engine v2.2.0, Copyright (c) 1998-2007 Zend Technologies
```

使用 php -m 查看PHP Modules

```
# bin/php -m
[PHP Modules]
cgi-fcgi
ctype
date
dom
filter
gd
hash
iconv
json
ldap
libxml
mssql
pcre
PDO
pdo_mysql
pdo_sqlite
posix
Reflection
session
SimpleXML
snmp
SPL
SQLite
standard
tokenizer
xml
xmlreader
xmlwriter
zip

[Zend Modules]
```

3.11.2.2. apt-get install

```
$ sudo apt-get install php5 php5-cli php5-cgi
```

参考php安装

找到 fastcgi.server 去掉注释

bin-path 改为PHP程序安装目录

```
fastcgi.server             = ( ".php" =>
                               ( "localhost" =>
                                 (
                                  "socket" => "/tmp/php-fastcgi.socket",
                                   "bin-path" => "/usr/local/php/bin/php"
                                 )
                               )
                            )
```

下面例子更复杂一些

1. /usr/local/lighttpd/etc/lighttpd.conf

```
include /usr/local/lighttpd/etc/php-fastcgi.conf
```

2. /usr/local/lighttpd/etc/php-fastcgi.conf

```
fastcgi.server = ( ".php" =>
        ( "localhost" =>
        ( "socket" => "/tmp/php-fastcgi.socket",
           "bin-path" => "/usr/local/php/bin/php",
           "min-procs" => 1,
           "max-procs" => 5,
           "max-load-per-proc" => 4,
           "idle-timeout" => 20
        )
        )
)
```

3. PHP FastCGI环境测试

echo "<?php phpinfo(); ?>" > /www/pages/index.php

curl http://127.0.0.1/index.php

3.11.3. Python

```
sudo apt-get install python
sudo apt-get install python-setuptools
```

3.11.3.1. Django

```
wget http://www.djangoproject.com/download/0.96/tarball/
tar zxvf Django-0.96.tar.gz
cd Django-0.96
python setup.py install
```

生成项目

```
django-admin.py startproject newtest
```

web server

```
cd newtest/
./manage.py runserver
```

helloworld.py

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("Hello, Django.")
```

urls.py

```
from django.conf.urls.defaults import *

urlpatterns = patterns('',
    # Example:
    # (r'^newtest/', include('newtest.foo.urls')),
    (r'^$', 'newtest.helloworld.index'),

    # Uncomment this for admin:
#     (r'^admin/', include('django.contrib.admin.urls')),
)
```

启动Web Server

```
# ./manage.py runserver
Validating models...
0 errors found.

Django version 0.96, using settings 'newtest.settings'
Development server is running at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

curl http://127.0.0.1:8000/

3.11.3.2. Python Imaging Library

Debian/Ubuntu

```
sudo apt-get install libjpeg62-dev
sudo apt-get install python-imaging
```

采用源码安装

```
tar zxvf Imaging-1.1.6.tar.gz
cd Imaging-1.1.6/
```

sudo python setup.py install,

decoder jpeg not available

首先确认jpeg库是否安装

find / -name jpeglib.h

然后修改头文件

Imaging-1.1.6/libImaging

修改Jpeg.h, #include "jpeglib.h" 改为

#include "/usr/include/jpeglib.h"

3.11.4. Perl

install fastcgi module

```
$ sudo apt-get install libfcgi-perl     libfcgi-procmanager-perl
```

3.11.4.1. Installing lighttpd and FastCGI for Catalyst

The examples also use a virtual host regexp that matches either www.myapp.com or myapp.com

```
$HTTP["host"] =~ "^(www.)?mysite.com"
```

Starting the FastCGI server

```
MyApp/script/myapp_fastcgi.pl -l /tmp/myapp.socket -n 5 -d
```

lighttpd.conf

```
server.document-root = "/var/www/MyApp/root"
```

$ sudo vim /etc/lighttpd/conf-available/10-fastcgi.conf

```
fastcgi.server = (
    "" => (
        "MyApp" => (
            "socket" => "/tmp/myapp.socket",
            "check-local" => "disable"
        )
    )
)
```

restart lighttpd

```
neo@master:~$ sudo /etc/init.d/lighttpd restart
 * Stopping web server lighttpd                            [ OK ]
 * Starting web server lighttpd                            [ OK ]
```

Testing

http://127.0.0.1/

More advanced configuration

例 3.2. fastcgi.conf

```
fastcgi.server = (
    "" => (
        "MyApp" => (
            "socket"       => "/tmp/myapp.socket",
            "check-local"  => "disable",
            "bin-path"     => "/var/www/MyApp/script/myapp_fastcgi.pl",
            "min-procs"    => 2,
            "max-procs"    => 5,
            "idle-timeout" => 20
        )
    )
)
```

3.11.5. Ruby

3.12. user-agent

```
$HTTP["user-agent"] =~ "Googlebot|Sosospider+|eMule|Wget|^Java|^PHP|Ruby|Python" {
  url.rewrite = ( "^/(.*)" => "/crawler.html" )
}
```

```
$HTTP["user-agent"] =~ "Baiduspider+" {
    connection.delay-seconds = 10
}
```

上一页 上一级 下一页

2. /etc/lighttpd/lighttpd.conf 起始页 4. 其他模块

4. 其他模块

4.1. mod_secdownload 防盗链

5. Example

5.1. s-maxage

s-maxage 头作用于反向代理服务器

例 3.3. Cache

```
$HTTP["url"] =~ "^/images/2010" {
        expire.url = ( "" => "access 15 minutes" )
}

$HTTP["host"] =~ "(img1|img2|img3)\.example\.com" {
    expire.url = ( "" => "access 15 minutes" )
    setenv.add-response-header  = ("Cache-Control" =>"s-maxage=3600")
}
```

第 4 章 Nginx

目录

1. Installing

1.1. Installing by apt-get under the debain/ubuntu

```
$ sudo apt-get install nginx
```

```
/etc/init.d/nginx start
```

1.2. CentOS

http://nginx.org/packages/centos/$releasever/$basearch/

$releasever 是版本号

$basearch 处理器架构

http://nginx.org/packages/centos/6/x86_64/

```
cat > /etc/yum.repos.d/nginx.repo <<EOF
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/6/x86_64/
gpgcheck=0
enabled=1
EOF
```

i386

```
cat > /etc/yum.repos.d/nginx.repo <<EOF
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/5/i386/
gpgcheck=0
enabled=1
EOF
```

```
yum search nginx
============================================= Matched: nginx
=============================================
nginx.x86_64 : high performance web server

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

1.3. installing by source

```
cd /usr/local/src/
wget http://www.nginx.org/download/nginx-1.0.6.tar.gz

./configure --prefix=/usr/local/server/nginx \
--with-openssl=/usr/include \
--with-pcre=/usr/include/pcre/ \
--with-http_stub_status_module \
--without-http_memcached_module \
--without-http_fastcgi_module \
--without-http_rewrite_module \
--without-http_map_module \
--without-http_geo_module \
--without-http_autoindex_module
```

rpm 所使用的编译参数

```
nginx -V
nginx: nginx version: nginx/1.0.6
nginx: built by gcc 4.4.4 20100726 (Red Hat 4.4.4-13) (GCC)
nginx: TLS SNI support enabled
nginx: configure arguments: --prefix=/etc/nginx/ --sbin-path=/usr/sbin/nginx --conf-
path=/etc/nginx/nginx.conf --error-log-path=/var/log/nginx/error.log --http-log-
path=/var/log/nginx/access.log --pid-path=/var/run/nginx.pid --lock-path=/var/run/nginx.lock --
http-client-body-temp-path=/var/cache/nginx/client_temp --http-proxy-temp-
path=/var/cache/nginx/proxy_temp --http-fastcgi-temp-path=/var/cache/nginx/fastcgi_temp --http-
uwsgi-temp-path=/var/cache/nginx/uwcgi_temp --http-scgi-temp-path=/var/cache/nginx/scgi_temp --
user=nginx --group=nginx --with-http_ssl_module --with-http_realip_module --with-
http_addition_module --with-http_sub_module --with-http_dav_module --with-http_flv_module --
with-http_gzip_static_module --with-http_random_index_module --with-http_secure_link_module --
with-http_stub_status_module --with-mail --with-mail_ssl_module --with-file-aio --with-ipv6
```

1.4. php-fpm

```
./configure --prefix=/srv/php-5.3.8 \
--with-config-file-path=/srv/php-5.3.8/etc \
--with-config-file-scan-dir=/srv/php-5.3.8/etc/conf.d \
--enable-fpm \
--with-fpm-user=www \
--with-fpm-group=www \
--with-pear \
--with-curl \
--with-gd \
--with-jpeg-dir \
--with-png-dir \
--with-freetype-dir \
--with-xpm-dir \
--with-iconv \
--with-mcrypt \
--with-mhash \
--with-zlib \
--with-xmlrpc \
--with-xsl \
--with-openssl \
--with-mysql=/srv/mysql-5.5.16-linux2.6-i686 \
--with-mysqli=/srv/mysql-5.5.16-linux2.6-i686/bin/mysql_config \
--with-pdo-mysql=/srv/mysql-5.5.16-linux2.6-i686 \
--with-sqlite=shared \
--with-pdo-sqlite=shared \
--disable-debug \
--enable-zip \
--enable-sockets \
--enable-soap \
--enable-mbstring \
--enable-magic-quotes \
--enable-inline-optimization \
--enable-gd-native-ttf \
--enable-xml \
--enable-ftp \
--enable-exif \
--enable-wddx \
--enable-bcmath \
--enable-calendar \
--enable-sqlite-utf8 \
--enable-shmop \
--enable-dba \
--enable-sysvsem \
--enable-sysvshm \
--enable-sysvmsg

make && make install
```

如果出现 fpm 编译错误，取消--with-mcrypt 可以编译成功。

```
# cp sapi/fpm/init.d.php-fpm /etc/init.d/php-fpm
# chmod 755 /etc/init.d/php-fpm
# ln -s /srv/php-5.3.5 /srv/php
# cp /srv/php/etc/php-fpm.conf.default /srv/php/etc/php-fpm.conf
# cp php.ini-production /srv/php/etc/php.ini
```

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

php-fpm.conf

```
# grep -v ';' /srv/php-5.3.5/etc/php-fpm.conf | grep -v "^$"
[global]
pid = run/php-fpm.pid
error_log = log/php-fpm.log
[www]
listen = 127.0.0.1:9000

user = www
group = www
pm = dynamic
pm.max_children = 2048
pm.start_servers = 20
pm.min_spare_servers = 5
pm.max_spare_servers = 35

pm.max_requests = 500
```

```
chkconfig --add php-fpm
```

1.5. rotate log

1.5.1. log shell

```

# cat /srv/bin/rotatelog.sh

#!/bin/bash
# run this script at 0:00

#Nginx Log Path
log_dir="/var/log/nginx"
date_dir=`date +%Y/%m/%d/%H`

mkdir -p ${log_dir}/${date_dir} > /dev/null 2>&1
mv ${log_dir}/access.log ${log_dir}/${date_dir}/access.log
mv ${log_dir}/error.log ${log_dir}/${date_dir}/error.log

kill -USR1 `cat /var/run/nginx.pid`

gzip ${log_dir}/${date_dir}/access.log &
gzip ${log_dir}/${date_dir}/error.log &
```

1.5.2. /etc/logrotate.d/nginx

```

# cat /etc/logrotate.d/nginx
/var/log/nginx/*.log {
        daily
        missingok
        rotate 52
        compress
        delaycompress
        notifempty
        create 640 root adm
        sharedscripts
        postrotate
                [ -f /var/run/nginx.pid ] && kill -USR1 `cat /var/run/nginx.pid`
        endscript
}
```

上一页 下一页

5. Example 起始页 2. fastcgi

2. fastcgi

2.1. spawn-fcgi

config php fastcgi

```
sudo vim /etc/nginx/sites-available/default

        location ~ \.php$ {
                fastcgi_pass   127.0.0.1:9000;
                fastcgi_index  index.php;
                fastcgi_param  SCRIPT_FILENAME  /scripts$fastcgi_script_name;
                include fastcgi_params;
        }
```

Spawn-fcgi

We still need a script to start our fast cgi processes. We will extract one from Lighttpd. and then disable start script of lighttpd

```
$ sudo apt-get install lighttpd
$ sudo chmod -x /etc/init.d/lighttpd
```

```
$ sudo touch /usr/bin/php-fastcgi
$ sudo vim /usr/bin/php-fastcgi

#!/bin/sh
/usr/bin/spawn-fcgi -a 127.0.0.1 -p 9000 -u www-data -f /usr/bin/php5-cgi
```

fastcgi daemon

```
$ sudo touch /etc/init.d/nginx-fastcgi
$ sudo chmod +x /usr/bin/php-fastcgi
$ sudo vim /etc/init.d/nginx-fastcgi

This is also a new empty file, add the following and save:

#!/bin/bash
PHP_SCRIPT=/usr/bin/php-fastcgi
RETVAL=0
case "$1" in
start)
$PHP_SCRIPT
RETVAL=$?
;;
stop)
killall -9 php
RETVAL=$?
;;
restart)
killall -9 php
$PHP_SCRIPT
RETVAL=$?
;;
*)
echo "Usage: nginx-fastcgi {start|stop|restart}"
exit 1
;;
esac
exit $RETVAL

We need to change some permissions to make this all work.
```

```
$ sudo chmod +x /etc/init.d/nginx-fastcgi
```

create a test file

```
sudo vim /var/www/nginx-default/index.php
<?php echo phpinfo(); ?>
```

2.2. php5-fpm

```
sudo apt-get install php5-fpm
```

3. worker_processes

worker_processes = CPU 数量

4. events

```
events {
    worker_connections  4096;
}
```

5. 可用的全局变量

```
$args
$content_length
$content_type
$document_root
$document_uri
$host
$http_user_agent
$http_cookie
$limit_rate
$request_body_file
$request_method
$remote_addr
$remote_port
$remote_user
$request_filename
$request_uri
$query_string
$scheme
$server_protocol
$server_addr
$server_name
$server_port
$uri
```

6. http 配置

6.1. X-Forwarded-For

```
real_ip_header X-Forwarded-For;
```

6.2. server

6.2.1. VirtualHost (虚拟主机)

```
# cat /etc/nginx/conf.d/images.conf
server {
    listen       80;
    server_name  images.example.com;

    #charset koi8-r;
    access_log  /var/log/nginx/images.access.log  main;

    location / {
        root   /www/images;
        index  index.html index.htm;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ \.php$ {
    #    proxy_pass   http://127.0.0.1;
    #}

    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
    #
    #location ~ \.php$ {
    #    root           html;
    #    fastcgi_pass   127.0.0.1:9000;
    #    fastcgi_index  index.php;
    #    fastcgi_param  SCRIPT_FILENAME  /scripts$fastcgi_script_name;
    #    include        fastcgi_params;
    #}

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny  all;
    #}
}
```

绑定多个域名

```
server_name  images.example.com img1.example.com img2.example.com;
```

使用通配符匹配

```
server_name  *.example.com
server_name  www.*;
```

正则匹配

```
server_name ~^(.+)\.example\.com$;
server_name ~^(www\.)?(.+)$;
```

6.2.2. location

```
    location / {
        root   /www;
        index  index.html index.htm;
    }
```

6.3. expires

```
#图片类资源缓存5天，并且不记录请求日志
location ~ .*\.(ico|gif|jpg|jpeg|png|bmp|swf)$
{
        expires      5d;
        access_log off;
}

#css/js 缓存一天，不记录请求日志
location ~ .*\.(js|css)?$
{
        expires      1d;
        access_log off;
}
```

```
location ~
.*\.(htm|html|gif|jpg|jpeg|png|bmp|swf|ioc|rar|zip|txt|flv|mid|doc|ppt|pdf|xls|mp3|wma)$
{
    expires      30d;
}
location ~ .*\.(js|css)?$
{
    expires      1h;
}
```

```
location ~* \.(js|css|jpg|jpeg|gif|png|swf)$ {
        if (-f $request_filename) {
           expires    1h;
           break;
        }
}

location ~ .*\.(gif|jpg|jpeg|png|bmp|swf|ico)$ {
    expires      30d;
    access_log   off;
}

location ~ .*\.(js|css)?$ {
    expires      30d;
    access_log   off;
}
```

6.4. access

```
#防止access文件被下载
location ~ /\.ht {
    deny  all;
}
```

```
location ~ ^/upload/.*\.php$
{
        deny all;
}

location ~ ^/static/images/.*\.php$
{
        deny all;
}
```

```
location ~ /\.ht {
    deny all;
}
```

```
location ~ .*\.(sqlite|sq3)$ {
    deny all;
}
```

6.5. autoindex

```
# vim /etc/nginx/sites-enabled/default

location  /  {
  autoindex  on;
}
```

```
# /etc/init.d/nginx reload
Reloading nginx configuration: nginx.
```

6.6. ssi

```
http {
  ssi  on;
}

location / {
  ssi on;
  ssi_silent_errors on;
  ssi_types text/shtml;
}
```

6.7. rewrite

```
Rewrite Flags
last - 基本上都用这个Flag。
break - 中止Rewirte，不在继续匹配
redirect - 返回临时重定向的HTTP状态302
permanent - 返回永久重定向的HTTP状态301

文件及目录匹配，其中:
-f和!-f用来判断是否存在文件
-d和!-d用来判断是否存在目录
-e和!-e用来判断是否存在文件或目录
-x和!-x用来判断文件是否可执行

正则表达式全部符号解释
~ 为区分大小写匹配
~* 为不区分大小写匹配
!~和!~* 分别为区分大小写不匹配及不区分大小写不匹配
(pattern) 匹配 pattern 并获取这一匹配。所获取的匹配可以从产生的 Matches 集合得到，在VBScript 中使用
SubMatches 集合，在JScript 中则使用 $0…$9 属性。要匹配圆括号字符，请使用 ‘\(’ 或 ‘\)’。
^ 匹配输入字符串的开始位置。
$ 匹配输入字符串的结束位置。
```

```
server {
        listen 80;
        server_name www.example.com example.com ;
        if ($host = "example.com" )
        {
                rewrite ^/(.*)$ http://www.example.com/$1 permanent;
        }
        if ($host != "www.example.com" )
        {
                rewrite ^/(.*)$ http://www.example.com/$1 permanent;
        }
}
```

```
location ~* \.(js|css|jpg|jpeg|gif|png|swf)$ {
        if (!-f $request_filename){
                rewrite /(.*) http://images.example.com/$1;
        }
}
```

```
if ($host ~ '(.*)\.static\.example\.com' ) {
    set $subdomain $1;
    rewrite  "^/(.*)$"  /$subdomain/$1;
}
```

6.8. gzip

```
gzip  on;
gzip_min_length  1000;
gzip_buffers     4 8k;
gzip_types       text/plain application/x-javascript text/css text/html application/xml;


gzip on;
gzip_http_version 1.0;
gzip_disable "MSIE [1-6].";
gzip_types text/plain application/x-javascript text/css text/javascript;
```

6.9. Cache

```

add_header     Nginx-Cache     "HIT  from  www.example.com";
or
add_header     Nginx-Cache     "$upstream_cache_status   from  www.example.com";

```

6.10. stub_status

```
location /nginx_status {
        stub_status on;
        access_log  on;
        allow 127.0.0.1;
        deny all;
}
```

6.11. server_tokens

```
http {
...
server_tokens off;
...
}
```

上一页	上一级	下一页
5. 可用的全局变量	起始页	7. Proxy

7. Proxy

```
# cat /etc/nginx/nginx.conf

#user  nobody;
worker_processes  4;

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

#pid        logs/nginx.pid;


events {
    worker_connections  40960;
        use epoll;
}


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

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    access_log  /dev/null;

    sendfile        on;
    #tcp_nopush     on;

    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;

upstream backend{
#        server 172.16.0.6:80;
        server 10.0.0.68:80;
        server 10.0.0.69:80;
}


    server {
        listen       80;
        server_name  localhost;

        #charset koi8-r;

        #access_log  logs/host.access.log  main;

#        location / {
#            root   html;
#            index  index.html index.htm;
#        }

    access_log  /dev/null;
    error_log   /dev/null;


    location / {
#        proxy_pass $scheme://$host$request_uri;
#        proxy_set_header Host $http_host;

#        proxy_buffers 256 4k;
#        proxy_max_temp_file_size 0;

#        proxy_connect_timeout 30;

#        proxy_cache_valid 200 302 10m;
#        proxy_cache_valid 301 1h;
#        proxy_cache_valid any 1m;



         proxy_pass      http://backend;

         proxy_redirect          off;
         proxy_set_header        Host $host;
#         proxy_set_header        X-Real-IP $remote_addr;
#         proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
```

```
         client_max_body_size    10m;
         client_body_buffer_size 128k;
         proxy_connect_timeout   30;
         proxy_send_timeout      30;
         proxy_read_timeout      30;
         proxy_buffer_size       4k;
         proxy_buffers           256 4k;
         proxy_busy_buffers_size 64k;
         proxy_temp_file_write_size 64k;
        tcp_nodelay on;
    }


        #error_page  404              /404.html;

        # redirect server error pages to the static page /50x.html
        #
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }
    }

}
```

7.1. request_filename + proxy_pass

如果文件不存在，那么去指定的节点上寻找

```
   location / {
        root  /www;
        proxy_intercept_errors  on;
        if (!-f $request_filename) {
          proxy_pass http://172.16.1.1;
          break;
        }
    }
        location / {
        root  /www/images;
        proxy_intercept_errors  on;
        if (!-f $request_filename) {
          proxy_pass http://172.16.1.2;
          break;
        }
    }
```

Home | Mirror | Search

Google™ Custom Search Search

第 5 章 Tomcat 安装与配置

目录

1. install java

解压安装

```
chmod +x jdk-6u1-linux-i586.bin
./jdk-6u1-linux-i586.bin
输入"yes"回车

mv jdk1.6.0_01 /usr/local/
ln -s /usr/local/jdk1.6.0_01/ /usr/local/java
```

/etc/profile.d/java.sh

例 5.1. /etc/profile.d/java.sh

```
################################################
### Java environment
################################################
export JAVA_HOME=/usr/local/java
export JRE_HOME=/usr/local/java/jre
export PATH=$PATH:/usr/local/java/bin:/usr/local/java/jre/bin
export CLASSPATH="./:/usr/local/java/lib:/usr/local/java/jre/lib:/usr/local/memcached/api/java"
export JAVA_OPTS="-Xms512m -Xmx1024m"
```

2. install tomcat

下载binary解压到/usr/local/

下载软件包

```
wget http://archive.apache.org/dist/tomcat/tomcat-6/v6.0.13/bin/apache-tomcat-6.0.13.tar.gz
wget http://archive.apache.org/dist/tomcat/tomcat-connectors/native/tomcat-native-1.1.10-src.tar.gz
wget http://archive.apache.org/dist/tomcat/tomcat-connectors/jk/source/jk-1.2.23/tomcat-connectors-1.2.23-src.tar.gz
```

```
tar zxvf apache-tomcat-6.0.13.tar.gz
mv apache-tomcat-6.0.13 /usr/local/
ln -s /usr/local/apache-tomcat-6.0.13/ /usr/local/tomcat
```

tomcat-native

```
tar zxvf tomcat-native-1.1.10-src.tar.gz
cd tomcat-native-1.1.10-src/jni/native
./configure --with-apr=/usr/local/apache/bin/apr-1-config --with-java-home=/usr/local/java/
make
make install
```

catalina.sh

```
CATALINA_OPTS="-Djava.library.path=/usr/local/apr/lib"
JAVA_OPTS="-Xss128k -Xms128m -Xmx1024m -XX:PermSize=128M -XX:MaxPermSize=256m -XX:MaxNewSize=256m"
```

启动

```
startup.sh
```

2.1. tomcat-native

```

cd /usr/local/tomcat-6.0.18/bin
tar zxvf tomcat-native.tar.gz
cd tomcat-native-1.1.14-src/jni/native
./configure --with-apr=/usr/local/apr --with-java-home=/usr/java/jdk1.6.0_11
make && make install
```

3. 配置 Tomcat 服务器

3.1. server.xml

```
    <Connector port="80" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
```

性能调整

```
    <Connector port="80" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443"
               maxThreads="2048" />

        <Connector port="80" protocol="HTTP/1.1"
                                maxThreads="2048"
                                minSpareThreads="64"
                                maxSpareThreads="256"
                                acceptCount="128"
                                enableLookups="false"
                                redirectPort="8443"
                                debug="0"
                                connectionTimeout="20000"
                                disableUploadTimeout="true"
                                URIEncoding="UTF-8" />
```

3.1.1. compression

压缩传送数据

```
compression="on"
compressionMinSize="2048"
noCompressionUserAgents="gozilla, traviata"
compressableMimeType="text/html,text/xml,text/plain,text/javascript,text/css"
```

3.1.2. useBodyEncodingForURI

如果你的站点编码非UTF-8,去掉URIEncoding="UTF-8"使用下面选项.

useBodyEncodingForURI="true"

3.1.3. HTTPS

```
   <Connector port="443" maxHttpHeaderSize="8192"
               maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
               enableLookups="false" disableUploadTimeout="true"
               acceptCount="100" scheme="https" secure="true"
               SSLEngine="on"
               SSLCertificateFile="${catalina.base}/conf/localhost.crt"
               SSLCertificateKeyFile="${catalina.base}/conf/localhost.key" />
```

3.1.4. 隐藏Tomcat版本信息

在Connector中加入server="Neo App Srv 1.0"

```
vim $CATALINA_HOME/conf/server.xml

    <Connector port="80" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443"
                                maxThreads="8192"
                                minSpareThreads="64"
                                maxSpareThreads="128"
                                acceptCount="128"
                                enableLookups="false"
                server="Neo App Srv 1.0"/>
```

```
# curl -I http://localhost:8080/
HTTP/1.1 400 Bad Request
Transfer-Encoding: chunked
Date: Thu, 20 Oct 2011 09:51:55 GMT
Connection: close
Server: Neo App Srv 1.0
```

3.1.5. vhost

传统配置方式

```
      <Host name="www.example.com"  appBase="webapps"
            unpackWARs="true" autoDeploy="true"
            xmlValidation="false" xmlNamespaceAware="false">
                        <Context path="" docBase="/www/example/www" debug="0"
reloadable="false"/>
      </Host>
      <Host name="news.example.com"  appBase="webapps"
            unpackWARs="true" autoDeploy="true"
            xmlValidation="false" xmlNamespaceAware="false">
                        <Context path="" docBase="/www/example/news" debug="0"
reloadable="false"/>
      </Host>
```

建议配置方式

```
vim server.xml

<Engine name="Catalina" defaultHost="neo">
    <Host name="neo"    appBase="neoapps"/>
    <Host name="other" appBase="otherapps"/>
</Engine>
```

Configuring Your Contexts

```
mkdir $CATALINA_HOME/conf/Catalina/neo

cp $CATALINA_HOME/conf/Catalina/localhost/manager.xml $CATALINA_HOME/conf/Catalina/neo/ROOT.xml

or

cp $CATALINA_HOME/conf/Catalina/localhost/manager.xml $CATALINA_HOME/conf/Catalina/neo
```

Webapps Directory

```
mkdir $CATALINA_HOME/neo
```

3.1.6. access_log

```
<Host name="localhost" ...>
  ...
  <Valve className="org.apache.catalina.valves.AccessLogValve"
         prefix="localhost_access_log." suffix=".txt"
         pattern="common"/>
  ...
</Host>
```

3.2. tomcat-users.xml

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>

<role rolename="manager"/>
<user username="tomcat" password="QI0Ajp7" roles="manager"/>

</tomcat-users>
```

状态监控 http://localhost/manager/status

服务管理 http://localhost/manager/html/list

3.3. logging.properties

修改日志目录

```
1catalina.org.apache.juli.FileHandler.level = FINE
#1catalina.org.apache.juli.FileHandler.directory = ${catalina.base}/logs
1catalina.org.apache.juli.FileHandler.directory = /www/logs/tomcat
1catalina.org.apache.juli.FileHandler.prefix = catalina.
```

Home | Mirror | Search

Google™ Custom Search Search

4. Connector

4.1. server.xml

vi conf/server.xml

```
    <Connector port="8009"
               maxThreads="4096"
               minSpareThreads="100"
               maxSpareThreads="500"
               enableLookups="false"
               acceptCount="15000"
               connectionTimeout="30000"
               redirectPort="8443"
               disableUploadTimeout="true"
               URIEncoding="UTF-8"
               protocol="AJP/1.3"/>
```

4.2. mod_jk

mod_jk 安装

```
tar zxvf tomcat-connectors-1.2.23-src.tar.gz
cd tomcat-connectors-1.2.23-src/native/
./configure --with-apxs=/usr/local/apache/bin/apxs
make
make install
chmod 755 /usr/local/apache/modules/mod_jk.so
```

httpd.conf 尾部加入

```
Include conf/mod_jk.conf
```

配置workers.properties

apache/conf/workers.properties

```
# Define 1 real worker using ajp13
worker.list=worker1
# Set properties for worker1 (ajp13)
worker.worker1.type=ajp13
worker.worker1.host=127.0.0.1
worker.worker1.port=8009
worker.worker1.lbfactor=1
worker.worker1.cachesize=128
worker.worker1.cache_timeout=600
worker.worker1.socket_keepalive=1
worker.worker1.reclycle_timeout=300
```

mod_jk.conf

apache/conf/mod_jk.conf

```
[chenjingfeng@d3010 Includes]$ cat mod_jk.conf
<IfModule mod_jk.c>
# Load mod_jk module
LoadModule jk_module            modules/mod_jk.so
# Where to find workers.properties
JkWorkersFile           /usr/local/apache/conf/workers.properties
# Where to put jk logs
JkLogFile               /usr/local/apache/logs/mod_jk.log
# Set the jk log level [debug/error/info]
JkLogLevel              error
# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
# JkOptions indicate to send SSL KEY SIZE,
JkOptions     +ForwardKeySize +ForwardURICompat -ForwardDirectories
# JkRequestLogFormat set the request format
JkRequestLogFormat     "%w %V %T"
JkShmFile     /usr/local/apache2/logs/mod_jk.shm
# Send jsp,servlet for context * to worker named worker1
JkMount  /status/* worker1
JkMount  /*.jsp worker1
JkMount  /*.jsps worker1
JkMount  /*.do worker1
JkMount  /*Servlet worker1
JkMount  /jk/* worker1
</IfModule>
```

分别测试apache,tomcat

4.3. mod_proxy_ajp

包含虚拟主机配置文件

# vi conf/httpd.conf

```
# Virtual hosts
Include conf/extra/httpd-vhosts.conf
```

虚拟主机中配置ProxyPass,ProxyPassReverse

# vi conf/extra/httpd-vhosts.conf

```
<VirtualHost *:80>
    ServerName netkiller.8800.org
    ProxyPass /images !
        ProxyPass /css !
        ProxyPass /js !
    ProxyPass /ajp ajp://localhost:8009/ajp
    ProxyPassReverse /ajp ajp://localhost:8009/ajp
</VirtualHost>
```

反向代理和均衡负载模块

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so

ProxyPass /admin balancer://tomcatcluster/admin lbmethod=byrequests stickysession=JSESSIONID
nofailover=Off timeout=5 maxattempts=3
ProxyPassReverse /admin balancer://tomcatcluster/admin

<Proxy balancer://tomcatcluster>
        BalancerMember ajp://localhost:8009 route=web1
        BalancerMember ajp://localhost:10009 smax=10 route=web2
        BalancerMember ajp://localhost:11009 route=web3
        BalancerMember ajp://localhost:12009 smax=10 route=web4
</Proxy>
```

4.4. RewriteEngine 连接 Tomcat

```
RewriteEngine On
```

```
RewriteRule ^/(.*) ajp://localhost:8009/ajp/$1 [P]
RewriteRule ^/(.*\.(jsp|do|sevlet)) ajp://localhost:8009/ajp/$1 [P]
```

4.5. Testing file

测试目录

```
[root@backup tomcat]# mkdir webapps/ajp
[root@backup tomcat]# mkdir webapps/jk
[root@backup tomcat]# vi webapps/ajp/index.jsp
[root@backup tomcat]# vi webapps/jk/index.jsp
```

测试文件

cat index.jsp

```
<%@ page contentType="text/html;charset=utf-8"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>apache+tomcat</title>
</head>

<body>
<%="It works!"%>
<%=new java.util.Date()%>
</body>
</html>
```

上一页	上一级	下一页
3. 配置 Tomcat 服务器	起始页	5. Init.d Script

5. Init.d Script

5.1. Script 1

```
#!/bin/bash
#############################################################
# Script for Apache and Tomcat
# File:/etc/rc.d/init.d/www
#############################################################
# Setup environment for script execution
#

# chkconfig: - 91 35
# description: Starts and stops the apache and tomcat daemons \
#               used to provide Neo Chen
#
# pidfile: /var/run/www/apache.pid
# pidfile: /var/run/www/tomcat.pid
# config:  /etc/apache2/apache2.conf


#APACHE_HOME=/usr/local/apache
#TOMCAT_HOME=/usr/local/tomcat
#APACHE_USER=apache
#TOMCAT_USER=tomcat

APACHE_HOME=/usr/local/apache-evaluation
TOMCAT_HOME=/usr/local/apache-tomcat-evaluation
APACHE_USER=root
TOMCAT_USER=root

OPEN_FILES=20480

# Source function library.
if [ -f /etc/init.d/functions ] ; then
  . /etc/init.d/functions
elif [ -f /etc/rc.d/init.d/functions ] ; then
  . /etc/rc.d/init.d/functions
else
  exit 0
fi

if [ ! -d /var/run/www ] ; then
  mkdir /var/run/www
fi

if [ -f /var/lock/subsys/tomcat ] ; then
        echo " "
fi

start() {
        if [ `ulimit -n` != ${OPEN_FILES} ]; then
                ulimit -n ${OPEN_FILES}
        fi
        echo -en "\\033[1;32;1m"
        echo "Starting Tomcat $TOMCAT_HOME ..."
        echo -en "\\033[0;39;1m"
        if [ -s /var/run/www/tomcat.pid ]; then
                echo "tomcat (pid `cat /var/run/www/tomcat.pid`) already running"
        else
                su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh start > /dev/null"
                echo `pgrep java` > /var/run/www/tomcat.pid
                touch /var/lock/subsys/tomcat
        fi
        sleep 2
        echo -en "\\033[1;32;1m"
        echo "Starting Apache $APACHE_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${APACHE_USER} -c "$APACHE_HOME/bin/apachectl start"
        touch /var/lock/subsys/apache
}

stop() {
        echo -en "\\033[1;32;1m"
        echo "Shutting down Apache $APACHE_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${APACHE_USER} -c "$APACHE_HOME/bin/apachectl stop"
        sleep 2
        echo -en "\\033[1;32;1m"
        echo "Shutting down Tomcat $TOMCAT_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh stop > /dev/null"
        rm -rf /var/run/www/tomcat.pid
        rm -f /var/lock/subsys/tomcat
        rm -f /var/lock/subsys/apache
}

restart() {
    stop
        if [ "`pgrep java`" = "" ]&& [ "`pgrep httpd`" = "" ]; then
                start
                exit 0
    else
                echo "Usage: $0 killall (^C)"
                echo -n "Waiting: "
    fi
    while true;
        do
                sleep 1
                if [ "`pgrep java`" = "" ] && [ "`pgrep httpd`" = "" ]; then
                        break
                else
                        echo -n "."
                        #echo -n "Enter your [y/n]: "; read ISKILL;
                fi
        done
        echo
    start
}

status() {
                ps -aux | grep -e tomcat -e apache

                echo -en "\\033[1;32;1m"
                echo ulimit open files: `ulimit -n`
                echo -en "\\033[0;39;1m"

                echo -en "\\033[1;32;1m"
                echo -en "httpd count:"
                ps axf|grep httpd|wc -l
                echo -en "\\033[0;39;1m"
}

killall() {
        if [ "`pgrep httpd`" != "" ]; then
                echo -en "\\033[1;32;1m"
                echo "kill Apache pid(`pgrep httpd`) ..."
                kill -9 `pgrep httpd`
                echo -en "\\033[0;39;1m"
        fi
        if [ "`pgrep java`" != "" ]; then
                echo -en "\\033[1;32;1m"
                echo "kill Tomcat pid(`pgrep java`) ..."
                kill -9 `pgrep java`
                echo -en "\\033[0;39;1m"
        fi
        rm -rf /var/run/www/tomcat.pid
        rm -f /var/lock/subsys/tomcat
        rm -f /var/lock/subsys/apache
}

# Determine and execute action based on command line parameter
case "$1" in
        start)
                start
                ;;
        stop)
                stop
                ;;
        restart)
                restart
                ;;
        status)
                status
                ;;
        killall)
                killall
                ;;
        *)
                echo -en "\\033[1;32;1m"
                echo "Usage: $1 {start|stop|restart|status|killall}"
                echo -en "\\033[0;39;1m"
                ;;
esac
echo -en "\\033[0;39;m"
exit 0
```

5.2. Shell Script 2

Apache,Tomcat 运行脚本

例 5.2. /etc/rc.d/init.d/www

```
#!/bin/bash
#############################################################
# Script for Apache and Tomcat
# File:/etc/rc.d/init.d/www
#############################################################
# Setup environment for script execution
#

# chkconfig: - 91 35
# description: Starts and stops the apache and tomcat daemons \
#              used to provide Neo Chen<openunix@163.com>
#
# pidfile: /var/run/www/apache.pid
# pidfile: /var/run/www/tomcat.pid
# config:  /etc/apache2/apache2.conf


#APACHE_HOME=/usr/local/apache
#TOMCAT_HOME=/usr/local/tomcat
#APACHE_USER=apache
#TOMCAT_USER=tomcat

APACHE_HOME=/usr/local/apache
TOMCAT_HOME=/usr/local/tomcat
APACHE_USER=root
TOMCAT_USER=root
WAIT_TIME=10
get_apache_pid(){
    APACHE_PID=`pgrep -o httpd`
    echo $APACHE_PID
}
get_tomcat_pid(){
    TOMCAT_PID=`ps axww | grep catalina.home | grep -v 'grep' | sed q | awk '{print $1}'`
    echo $TOMCAT_PID
}

#OPEN_FILS=40960

# Source function library.
#if [ -f /etc/init.d/functions ] ; then
#  . /etc/init.d/functions
#elif [ -f /etc/rc.d/init.d/functions ] ; then
#  . /etc/rc.d/init.d/functions
#else
#  exit 0
#fi

if [ ! -d /var/run/www ] ; then
  mkdir /var/run/www
fi

#if [ -f /var/lock/subsys/tomcat ] ; then
#fi

start() {
        #if [ `ulimit -n` -le  ${OPEN_FILES} ]; then
        #       ulimit -n ${OPEN_FILES}
        #fi
        echo -en "\\033[1;32;1m"
        echo "Starting Tomcat $TOMCAT_HOME ..."
        echo -en "\\033[0;39;1m"
        if [ -s /var/run/www/tomcat.pid ]; then
                echo "tomcat (pid `cat /var/run/www/tomcat.pid`) already running"
        else
                su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh start > /dev/null"
                echo `get_tomcat_pid` > /var/run/www/tomcat.pid
                touch /var/lock/subsys/tomcat
        fi
        sleep 2
        echo -en "\\033[1;32;1m"
        echo "Starting Apache $APACHE_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${APACHE_USER} -c "$APACHE_HOME/bin/apachectl start"
        touch /var/lock/subsys/apache
}

stop() {
        echo -en "\\033[1;32;1m"
        echo "Shutting down Apache $APACHE_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${APACHE_USER} -c "$APACHE_HOME/bin/apachectl stop"
        sleep 2
        echo -en "\\033[1;32;1m"
        echo "Shutting down Tomcat $TOMCAT_HOME ..."
        echo -en "\\033[0;39;1m"
        su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh stop > /dev/null"
        rm -rf /var/run/www/tomcat.pid
        rm -f /var/lock/subsys/tomcat
        rm -f /var/lock/subsys/apache
}

restart() {
    stop
    sleep 2
    if [ -z `get_tomcat_pid` ]&& [ -z `get_apache_pid` ]; then
                start
                exit 0
    else
                echo "Usage: $0 killall (^C)"
                echo -n "Waiting: "
    fi
    while true;
        do
                sleep 1
                if [ -z `get_tomcat_pid` ] && [ -z `get_apache_pid` ]; then
```

```
                        break
                else
                        echo -n "."
                fi
        done
        echo
    start
}

k9restart() {
    ISEXIT='false'
    stop
    for i in `seq 1 ${WAIT_TIME}`;
    do
                if [ -z `get_tomcat_pid` ] && [ -z `get_apache_pid` ]; then
                ISEXIT='true'
                break
                else
                        sleep 1
                fi
        done

        if [ $ISEXIT == 'false' ]; then
            while true;
                do
                        if [ -z `get_tomcat_pid` ] && [ -z `get_apache_pid` ]; then
                                ISEXIT='true'
                        break
                        fi

                        if [ -n `get_apache_pid` ]; then
                                kill -9 `pgrep httpd`
                        fi
                        if [ -n `get_tomcat_pid` ]; then
                                kill -9 `get_tomcat_pid`
                        fi
                done
                rm -rf /var/run/www/tomcat.pid
                rm -f /var/lock/subsys/tomcat
                rm -f /var/lock/subsys/apache
        fi

        echo

        if [ $ISEXIT == 'true' ]; then
                start
        fi
}

status() {
                #ps -aux | grep -e tomcat -e apache

                echo -en "\\033[1;32;1m"
                echo ulimit open files: `ulimit -n`
                echo -en "\\033[0;39;1m"

                echo -en "\\033[1;32;1m"
                echo -en "httpd count:"
                let hc=`ps axf|grep httpd|wc -l`-1
                echo $hc
                echo -en "apache count:"
                netstat -alp | grep '*:http' | wc -l
                echo -en "tomcat count:"
                netstat -alp | grep '*:webcache' | wc -l
                echo -en "dbconn count:"
                netstat -a | grep ':3433' | wc -l
                echo -en "\\033[0;39;1m"
}

kall() {
        if [ `get_apache_pid` ]; then
                echo -en "\\033[1;32;1m"
                echo "kill Apache pid(`pgrep httpd`) ..."
                kill `pgrep httpd`
                echo -en "\\033[0;39;1m"
        fi
        if [ `get_tomcat_pid` ]; then
                echo -en "\\033[1;32;1m"
                echo "kill Tomcat pid(`pgrep java`) ..."
                kill `pgrep java`
                echo -en "\\033[0;39;1m"
        fi
        rm -rf /var/run/www/tomcat.pid
        rm -f /var/lock/subsys/tomcat
        rm -f /var/lock/subsys/apache
}

reload() {
        killall -HUP httpd
}

tomcat_restart() {
    su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh stop > /dev/null"
    rm -rf /var/run/www/tomcat.pid
    rm -f /var/lock/subsys/tomcat
    sleep 2
    if [ -z `get_tomcat_pid` ]; then
        su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh start > /dev/null"
        exit 0
    else
        echo "Usage: $0 killall (^C)"
        echo -n "Waiting: "
    fi
    while true;
    do
```

```
                sleep 1
                if [ -z `get_tomcat_pid` ]; then
                        echo
                        break
                else
                        echo -n "."
                        #echo -n "Enter your [y/n]: "; read ISKILL;
                fi
    done
    su - ${TOMCAT_USER} -c "$TOMCAT_HOME/bin/catalina.sh start > /dev/null"
    echo `get_tomcat_pid` > /var/run/www/tomcat.pid
    touch /var/lock/subsys/tomcat
}


# Determine and execute action based on command line parameter
case $1 in
    apache)
        case "$2" in
            reload)
                reload
                ;;
            *)
                su - ${APACHE_USER} -c "${APACHE_HOME}/bin/apachectl $2"
                ;;
        esac
        ;;
    tomcat)
        case "$2" in
            restart)
                tomcat_restart
                ;;
            *)
                su - ${TOMCAT_USER} -c "${TOMCAT_HOME}/bin/catalina.sh $2"
                ;;
        esac
        ;;
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        restart
        ;;
    status)
        status
        ;;
    killall)
        kall
        ;;
    k9restart)
        k9restart >/dev/null
        ;;
    *)
        echo -en "\\033[1;32;1m"
        echo "Usage: $0 {start|stop|restart|status|killall|k9restart}"
        echo "Usage: $0 apache {start|restart|graceful|graceful-stop|stop|reload}"
        echo "Usage: $0 tomcat {debug|run|start|restart|stop|version}"
        echo -en "\\033[0;39;1m"
        ;;
esac
echo -en "\\033[0;39;m"
exit 0
```

```
chmod 700 /etc/init.d/www
```

第 6 章 Resin

目录

http://www.caucho.com

1. 安装Resin

JRE

```
$ sudo apt-get install sun-java6-jre
```

下载Resin

注意: Resin Pro 与 Resin 前者要Licence

1.1. 直接使用

简易安装，直接解压缩后即可使用

```
$ wget http://www.caucho.com/download/resin-4.0.1.tar.gz
$ tar zxvf resin-4.0.1.tar.gz
$ sudo mv resin-4.0.1 ..
$ cd ..
$ sudo ln -s resin-4.0.1 resin
```

1.2. Debian/Ubuntu

```
$ wget http://www.caucho.com/download/resin_4.0.1-i386.deb
```

安装 Resin

```
$ sudo dpkg -i resin_4.0.1-i386.deb
```

1.3. 源码安装Resin

源码安装

```

$ cd /usr/local/src/
$ wget http://www.caucho.com/download/resin-4.0.1.tar.gz
$ tar zxvf resin-4.0.1.tar.gz
$ ./configure --prefix=/usr/local/resin-4.0.1 \
--with-apxs=/usr/local/httpd/bin/apxs \
--with-java-home=/usr/local/java \
--enable-64bit \
--enable-lfs \
--enable-ssl \
--enable-debug
$ make && make install
$ cd ..
$ sudo ln -s resin-4.0.1 resin
```

设置 resin 以服务的形式开机自启动

```
$ sudo cp /usr/local/resin/contrib/init.resin /etc/init.d/resin
$ sudo chmod 755 /etc/init.d/resin
$ sudo update-rc.d resin defaults 99
```

2. Compiling mod_caucho.so

```
unix> ./configure --with-apxs=/usr/local/apache/bin/apxs
unix> make && make install
```

```
#
# mod_caucho Resin Configuration
#
LoadModule caucho_module /usr/local/apache/modules/mod_caucho.so
ResinConfigServer localhost 6802
CauchoConfigCacheDirectory /tmp
CauchoStatus yes
<Location /caucho-status>
  SetHandler caucho-status
</Location>
```

```
<IfModule mod_caucho.c>
ResinConfigServer localhost 6802
<Location /caucho-status>
SetHandler caucho-status
</Location>
</IfModule>

AddHandler caucho-request jsp
<Location /servlet/*>
SetHandler caucho-request
</Location>


<IfModule mod_caucho.c>
        <LocationMatch (.*?)\.action>
                SetHandler caucho-request
        </LocationMatch>
        <LocationMatch (.*?)\.jsp>
                SetHandler caucho-request
        </LocationMatch>
        <LocationMatch (.*?)\.do>
                SetHandler caucho-request
        </LocationMatch>
</IfModule>
```

3. resin.conf

3.1. Maximum number of threads

Maximum number of threads.

```
<thread-max>4096</thread-max>
```

thread-max数值需要使用ab命令做压力测试，逐步调整。

3.2. Configures the keepalive

```
      <!-- Configures the keepalive -->
      <keepalive-max>128</keepalive-max>
      <keepalive-timeout>15s</keepalive-timeout>
```

3.3. ssl

```
<http address="*" port="443">
  <openssl>
  <certificate-file>/srv/keys/example.com/star.example.com.crt</certificate-file>
  <certificate-key-file>/srv/keys/example.com/star.example.com.key</certificate-key-file>
  <password>4fff74da-aea4-a9fc-4b5f-e6d497588726</password>
  </openssl>
</http>
```

自颁发证书，首先是使用keytool工具安装证书

```
生成证书:
keytool –genkeypair –keyalg RSA –keysize 2048 SHA1withRSA –validity 3650  -alias neo –keystore
server.keystore –storepass password –dname "CN=www.example.com, OU=test, O=example.com, L=SZ,
ST=GD, C=CN"

导出证书
-keytool –exportcert –alias neo –keystore server.keystore –storepass password  –file server.cer
–rfc

打印证书
Keytool -printcert –file server.cer

导出证书签发申请
Keytool –certreg –aias neo –keystore server.keystore –storepass password –file ins.csr –v

导入证书
Keytool –importcert –trustcacerts –alias neo –file server.cer –keystore server.keystore –
storepass password

查看数字证书
Keytool -list

当成功的导入了证书以后就要容器中进行配置才可以使用
首先是要把证书中的那个 server.keystore 和 server.cer这两个文件放入到Resin服务器的keys这个文件夹中 如果没有的
话 就手动的建立这个文件夹
然后去 config 文件夹下配置你的配置文件
我在resin 这个容器中的配置如下
```

```
<http address="*" port="443">
    <jsse-ssl>
       <key-store-file>keys/server.keystore</key-store-file>
       <password>password</password>
    </jsse-ssl>
</http>
```

上一页	上一级	下一页
2. Compiling mod_caucho.so	起始页	4. virtual hosts

4. virtual hosts

4.1. explicit host

例 6.1. explicit host in resin.conf

```
<resin xmlns="http://caucho.com/ns/resin">
<cluster id="">

<host host-name="www.foo.com">
  <host-alias>foo.com</host-alias>
  <host-alias>web.foo.com</host-alias>

  <root-directory>/opt/www/www.foo.com</root-directory>

  <web-app id="/" document-directory="webapps/ROOT">

  </web-app>
  ...
</host>

</cluster>
</resin>
```

4.2. regexp host

例 6.2. regexp host in resin.conf

```
<resin xmlns="http://caucho.com/ns/resin">
<cluster id="">

<host regexp="([^.]+)\.foo\.com">
  <host-name>${host.regexp[1]}.foo.com</host-name>

  <root-directory>/var/www/hosts/www.${host.regexp[1]}.com</root-directory>

  ...
</host>

</cluster>
</resin>
```

4.3. host-alias

例 6.3. host-alias in the resin.conf

```
<resin xmlns="http://caucho.com">
<cluster id="">
```

```
  <host id="www.foo.com" root-directory="/var/www/foo.com">
    <host-alias>foo.com</host-alias>

    <web-app id=""/>
  </host>

</cluster>
</resin>
```

例 6.4. host-alias in a /var/www/hosts/foo/host.xml

```
<host xmlns="http://caucho.com">

  <host-name>www.foo.com</host-name>
  <host-alias>foo.com</host-alias>

  <web-app id="" root-directory="htdocs"/>

</host>
```

例 6.5. host-alias-regexp in the resin.conf

```
<resin xmlns="http://caucho.com">
<cluster id="">

  <host id="www.foo.com" root-directory="/var/www/foo.com">
    <host-alias-regexp>.*foo.com</host-alias-regexp>

    <web-app id=""/>
  </host>

</cluster>
</resin>
```

4.4. configures a deployment directory for virtual hosts

```
<resin xmlns="http://caucho.com/ns/resin">
  <cluster id="app-tier">
    <root-directory>/var/www</root-directory>

    <host-deploy path="hosts">
      <host-default>
        <resin:import path="host.xml" optional="true"/>

        <web-app-deploy path="webapps"/>
      </host-default>
    </host-deploy>
  </cluster>
</resin>
```

$RESIN_HOME/hosts其下的任何目录将对应一个虚拟主机。在$RESIN_HOME/hosts下也可以放置jar文件，其会被展开变成一个虚拟主机。

```
$RESIN_HOME/hosts/www.example.com
$RESIN_HOME/hosts/www.example.net
$RESIN_HOME/hosts/www.example.org
```

4.5. Resources

例 6.6. shared database in host

```
<resin xmlns="http://caucho.com/ns/resin">
  <cluster id="app-tier">
     <server id="a" .../>

     <host id="www.foo.com">
        <database jndi-name="jdbc/test">
            <driver type="org.postgresql.Driver">
                <url>jdbc:postgresql://localhost/test</url>
                <user>caucho</user>
            </driver>
        </database>

        <web-app-default path="webapps"/>
    </host>
  </cluster>
</resin>
```

Oracle JDBC

```
<database>
        <jndi-name>jdbc/test</jndi-name>
        <driver type="oracle.jdbc.pool.OracleConnectionPoolDataSource">
        <url>jdbc:oracle:thin:@172.16.0.1:1521:database</url>
        <user>user</user>
        <password>password</password>
        </driver>
        <prepared-statement-cache-size>8</prepared-statement-cache-size>
        <max-connections>1024</max-connections>
        <max-idle-time>20s</max-idle-time>
</database>
```

例 6.7. rewrite-dispatch

```
<resin xmlns="http://caucho.com/ns/resin">
  <cluster id="app-tier">

    <host host-name="www.foo.com">
      <rewrite-dispatch>
        <redirect regexp="^/foo" target="/index.php?foo="/>
      </rewrite-dispatch>
    </host>

  </cluster>
</resin>
```

上一页 上一级 下一页

3. resin.conf 起始页 5. FAQ

5. FAQ

5.1. java.lang.OutOfMemoryError: PermGen space

```
vim /usr/local/resin/conf/resin.conf

<jvm-arg>-XX:PermSize=128M</jvm-arg>
<jvm-arg>-XX:MaxPermSize=512m</jvm-arg>
```

第 7 章 Application Server

目录

1. Zope

参考Python安装

1. 下载 Zope-3

```
wget http://www.zope.org/Products/Zope3/3.3.1/Zope-3.3.1.tgz
tar zxvf Zope-3.3.1.tgz
cd cd Zope-3.3.1
```

2. configure

```
./configure --prefix=/usr/local/Zope --with-python=/usr/local/python2.4/bin/python

make
make check
make install
```

3. 创建一个Zope实例

```
cd /usr/local/Zope
./bin/mkzopeinstance -u neo:chen -d /usr/local/Zope/webapps
cd webapps
./bin/runzope
```

4. 测试

```
http://netkiller.8800.org:8080/
```

2. JBoss - JBoss Enterprise Middleware

参考Java安装

1. 下载安装 JBoss

```
cd /usr/local/src/
wget http://nchc.dl.sourceforge.net/sourceforge/jboss/jboss-5.0.0.Beta2.zip
unzip jboss-5.0.0.Beta2.zip
mv jboss-5.0.0.Beta2 ..
cd ..
ln -s jboss-5.0.0.Beta2 jboss
```

2. 运行 Jboss

```
cd jboss/bin
chmod +x *.sh
./run.sh
```

第 8 章 Search Engine

目录

1. Solr

http://lucene.apache.org/solr/

java 采用apt-get安装

例 8.1. /etc/profile.d/java.sh

```
################################################
### Java environment by neo
################################################
export JAVA_HOME=/usr
export JRE_HOME=/usr
export PATH=$PATH:/usr/local/apache-tomcat/bin/:/usr/local/jetty-6.1.18/bin
export CLASSPATH="./:/usr/share/java/:/usr/local/apache-solr/example/multicore/lib"
export JAVA_OPTS="-Xms128m -Xmx1024m"
```

1.1. Embedded Jetty

```
wget http://apache.freelamp.com/lucene/solr/1.3.0/apache-solr-1.3.0.tgz
tar zxvf apache-solr-1.3.0.tgz
ln -s apache-solr-1.3.0 ../apache-solr
cd ../apache-solr/example/
java -jar start.jar
```

multicore: java -Dsolr.solr.home=multicore -jar start.jar

1.2. Jetty

http://jetty.mortbay.org/jetty/

过程 8.1. apt-get install

1. install

```
$ sudo apt-get install libxpp3-java
$ sudo apt-get install solr-jetty
```

2. firewall

```
$ sudo ufw allow 8280
```

3. Testing.

   http://172.16.0.1:8280/

   http://172.16.0.1:8280/admin/ (user:admin, passwd:admin)

过程 8.2. source codes install

- download

```
wget http://dist.codehaus.org/jetty/jetty-6.1.18/jetty-6.1.18.zip
```

1.3. Tomcat

http://tomcat.apache.org/

1. download

```
cd /usr/local/src

wget http://apache.etoak.com/tomcat/tomcat-6/v6.0.20/bin/apache-tomcat-6.0.20.tar.gz
wget http://apache.freelamp.com/lucene/solr/1.3.0/apache-solr-1.3.0.tgz

tar zxvf apache-tomcat-6.0.20.tar.gz
ln -s apache-tomcat-6.0.20 ../apache-tomcat

tar zxvf apache-solr-1.3.0.tgz
ln -s apache-solr-1.3.0 ../apache-solr
```

2. solr.xml

```
vim /usr/local/apache-tomcat/conf/Catalina/localhost/solr.xml

<Context docBase="/usr/local/apache-solr/dist/apache-solr-1.3.0.war" debug="0"
crossContext="true" >
   <Environment name="solr/home" type="java.lang.String" value="/usr/local/apache-
solr/example/solr" override="true" />
</Context>
```

1.4. solr-php-client

http://code.google.com/p/solr-php-client/

```
wget http://solr-php-client.googlecode.com/files/SolrPhpClient.2009-03-11.tgz
tar zxvf SolrPhpClient.2009-03-11.tgz
sudo mv SolrPhpClient/Apache /usr/share/php/
```

1.5. multicore

solr.xml

```
vim /usr/local/apache-solr/example/multicore/solr.xml

<?xml version="1.0" encoding="UTF-8" ?>
<solr persistent="false">
  <cores adminPath="/admin/cores">
    <core name="core0" instanceDir="core0" />
    <core name="core1" instanceDir="core1" />

    <core name="article" instanceDir="article" />

  </cores>
</solr>
```

core directory and config file

```
mkdir -p article/conf

vim article/conf/solrconfig.xml

<?xml version="1.0" encoding="UTF-8" ?>
<config>
  <updateHandler class="solr.DirectUpdateHandler2" />
  <requestDispatcher handleSelect="true" >
    <requestParsers enableRemoteStreaming="false" multipartUploadLimitInKB="2048" />
  </requestDispatcher>
  <requestHandler name="standard" class="solr.StandardRequestHandler" default="true" />
  <requestHandler name="/update" class="solr.XmlUpdateRequestHandler" />
  <requestHandler name="/admin/" class="org.apache.solr.handler.admin.AdminHandlers" />
  <admin>
    <defaultQuery>solr</defaultQuery>
  </admin>
</config>

vim article/conf/schema.xml

<?xml version="1.0" ?>
<schema name="example core zero" version="1.1">
  <types>
   <fieldType name="sint" class="solr.SortableIntField" sortMissingLast="true"
```

```
omitNorms="true"/>
   <fieldtype name="string"  class="solr.StrField" sortMissingLast="true" omitNorms="true"/>
   <fieldType name="date" class="solr.DateField" sortMissingLast="true" omitNorms="true"/>
   <fieldType name="text" class="solr.TextField" positionIncrementGap="100" />
  </types>
 <fields>
  <!-- general -->
   <field name="id"      type="sint"   indexed="true"  stored="true"  multiValued="false"
required="true"/>
  <field name="type"    type="string"   indexed="true"  stored="true"  multiValued="false" />
  <field name="name"    type="string"   indexed="true"  stored="true"  multiValued="false" />
  <field name="title"   type="string"   indexed="true"  stored="true"  multiValued="false" />
  <field name="content"   type="text"   indexed="true"  stored="true"  multiValued="false" />
  <field name="timestamp" type="date" indexed="true" stored="true" default="NOW"/>
 </fields>
 <!-- field to use to determine and enforce document uniqueness. -->
 <uniqueKey>id</uniqueKey>
 <!-- field for the QueryParser to use when an explicit fieldname is absent -->
 <defaultSearchField>content</defaultSearchField>
 <!-- SolrQueryParser configuration: defaultOperator="AND|OR" -->
 <solrQueryParser defaultOperator="OR"/>
        <copyField source="title" dest="content"/>
        <copyField source="name" dest="content"/>
</schema>
```

commit datas

```
vim test.xml

<add>
        <doc>
          <field name="id">1</field>
          <field name="name">Hello world</field>
        </doc>

        <doc>
          <field name="id">2</field>
          <field name="title">Title Hello world</field>
        </doc>

        <doc>
          <field name="id">3</field>
          <field name="name">Hello world 1</field>
          <field name="content">Content 1</field>
        </doc>

        <doc>
          <field name="id">4</field>
          <field name="name">Name Neo</field>
        </doc>

        <doc>
          <field name="id">5</field>
          <field name="name">Last Chan</field>
        </doc>
</add>


java -Durl=http://localhost:8983/solr/article/update -Dcommit=yes -jar ../exampledocs/post.jar
test.xml
```

1.6. 中文分词

1.6.1. ChineseTokenizerFactory

```
<fieldType name="text" class="solr.TextField" >
    <analyzer>
        <tokenizer class="org.apache.solr.analysis.ChineseTokenizerFactory"/>
    </analyzer>
</fieldType>
```

1.6.2. CJK

```
<fieldType name="text" class="solr.TextField" positionIncrementGap="100">
        <analyzer>
        <tokenizer class="solr.CJKTokenizerFactory"/>
        </analyzer>
</fieldType>
```

1.6.3. mmseg4j

http://code.google.com/p/mmseg4j/

install

```
$ cd /usr/local/src/
$ wget http://mmseg4j.googlecode.com/files/mmseg4j-1.7.2.zip
$ unzip mmseg4j-1.7.2.zip
$ mkdir /usr/local/apache-solr/example/multicore/lib
$ cp /usr/local/src/mmseg4j-1.7.2/mmseg4j-all-1.7.2.jar /usr/local/apache-
solr/example/multicore/lib
$ cd mmseg4j-1.7.2/
```

test

```
$ java -Dmmseg.dic.path=/usr/local/apache-solr/example/solr -jar mmseg4j-all-1.7.2.jar 这里是字符
串
$ java -Dmmseg.dic.path=/usr/local/apache-solr/example/solr -cp .:mmseg4j-all-1.7.2.jar
com.chenlb.mmseg4j.example.Simple 这里是字符串
$ java -Dmmseg.dic.path=/usr/local/apache-solr/example/solr -cp .:mmseg4j-all-1.7.2.jar
com.chenlb.mmseg4j.example.MaxWord 这里是字符串
```

mmseg4j 在 solr 中主要支持两个参数：mode、dicPath。mode 表示是什么模式分词（有效值：simplex、complex、max-word，如果输入了无效的默认用 max-word。）。dicPath 是词库目录可以是绝对目录，也可以是相对目录（是相对 solr.home 目录下的，dic 就会在 solr.home/dic 目录下找词库文件），如果不指定就是默认在 CWD/data 目录（程序运行当前目录的data子目录）下找。

分词例子

```
<fieldtype name="textComplex" class="solr.TextField">
        <analyzer>
                <tokenizer class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory"
                        mode="complex" dicPath="dic">
                </tokenizer>
        </analyzer>
</fieldtype>

<fieldtype name="textMaxWord" class="solr.TextField">
        <analyzer>
                <tokenizer class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory"
                        mode="max-word" dicPath="dic">
                </tokenizer>
        </analyzer>
</fieldtype>

<fieldtype name="textSimple" class="solr.TextField">
        <analyzer>
                <tokenizer class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory"
                        mode="simple" dicPath="/usr/local/apache-solr/example/solr/my_dic">
                </tokenizer>
        </analyzer>
</fieldtype>
```

添加到schema.xml

```
<fieldType name="text" class="solr.TextField" positionIncrementGap="100" >
    <analyzer>
        <tokenizer class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory" mode="complex"
dicPath="dic"/>
        <filter class="solr.LowerCaseFilterFactory"/>
    </analyzer>
</fieldType>
```

http://localhost:8080/solr/admin/analysis.jsp 在 Field 的下拉菜单选择 name，然后在应用输入 complex。可以看 mmseg4j 的分词的结果.

1.6.4. 中文分词“庖丁解牛” Paoding Analysis

```
$ cd /usr/local/src/
$ mkdir paoding-analysis-2.0.4-beta
$ cd paoding-analysis-2.0.4-beta/
$ wget http://paoding.googlecode.com/files/paoding-analysis-2.0.4-beta.zip
$ unzip paoding-analysis-2.0.4-beta.zip
$ cp paoding-analysis.jar /usr/local/apache-solr/example/multicore/lib/
```

ChineseTokenizerFactory

上一页 下一页

2. JBoss - JBoss Enterprise Middleware 起始页 2. Nutch

2. Nutch

http://lucene.apache.org/nutch/

How to Setup Nutch and Hadoop

http://wiki.apache.org/nutch/NutchHadoopTutorial

1. 下载

```
$ cd /usr/local/src/
$ wget http://apache.etoak.com/lucene/nutch/nutch-1.0.tar.gz
$ tar zxvf nutch-1.0.tar.gz
$ sudo cp -r nutch-1.0 ..
$ cd ..
$ sudo ln -s nutch-1.0 apache-nutch
```

2. 创建文件myurl

```
$ cd apache-nutch
$ mkdir urls
$ vim urls/myurl
http://netkiller.8800.org/
```

3. 配置文件 crawl-urlfilter.txt

   编辑conf/crawl-urlfilter.txt文件，修改MY.DOMAIN.NAME部分，把它替换为你想要抓取的域名

```
$ cp conf/crawl-urlfilter.txt conf/crawl-urlfilter.txt.old
$ vim conf/crawl-urlfilter.txt

# accept hosts in MY.DOMAIN.NAME
+^http://([a-z0-9]*\.)*MY.DOMAIN.NAME/
修改为:
# accept hosts in MY.DOMAIN.NAME
+^http://([a-z0-9]*\.)*netkiller.8800.org/
```

4. http.agent.name

```

$ vim conf/nutch-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
  <name>http.agent.name</name>
  <value>Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1) Gecko/20090624
Firefox/3.5</value>
  <description>HTTP 'User-Agent' request header. MUST NOT be empty -
  please set this to a single word uniquely related to your organization.

  NOTE: You should also check other related properties:

    http.robots.agents
    http.agent.description
    http.agent.url
    http.agent.email
    http.agent.version

  and set their values appropriately.

  </description>
</property>

<property>
  <name>http.agent.description</name>
  <value></value>
  <description>Further description of our bot- this text is used in
  the User-Agent header.  It appears in parenthesis after the agent name.
  </description>
</property>

<property>
  <name>http.agent.url</name>
  <value>http://netkiller.8800.org/robot.html</value>
  <description>A URL to advertise in the User-Agent header.  This will
   appear in parenthesis after the agent name. Custom dictates that this
   should be a URL of a page explaining the purpose and behavior of this
   crawler.
  </description>
</property>

<property>
  <name>http.agent.email</name>
  <value>openunix@163.com</value>
  <description>An email address to advertise in the HTTP 'From' request
   header and User-Agent header. A good practice is to mangle this
   address (e.g. 'info at example dot com') to avoid spamming.
  </description>
</property>

</configuration>
```

5. 运行以下命令行开始工作

$ bin/nutch crawl urls -dir crawl -depth 3 -threads 5

```
bin/nutch crawl <your_url> -dir <your_dir> -depth 2 -threads 4 >&logs/logs1.log

urls 存放需要爬行的url文件的目录，即目录/nutch/urls。
-dir  dirnames          设置保存所抓取网页的目录.
-depth  depth           表明抓取网页的层次深度
-delay  delay           表明访问不同主机的延时，单位为“秒”
-threads  threads       表明需要启动的线程数
-topN 50        topN    一个网站保存的最大页面数。


$ nohup bin/nutch crawl /usr/local/apache-nutch/urls -dir /usr/local/apache-nutch/crawl -
depth 5 -threads 50 -topN 50 > /tmp/nutch.log &
```

6. depoly

```
$ cd /usr/local/apache-tomcat/conf/Catalina/localhost
$ vim nutch.xml
<Context docBase="/usr/local/apache-nutch/nutch-1.0.war" debug="0" crossContext="true" >
</Context>
```

searcher.dir

```
$ vim /usr/local/apache-tomcat/webapps/nutch/WEB-INF/classes/nutch-site.xml

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>searcher.dir</name>
        <value>/usr/local/apache-nutch/crawl</value>
    </property>
</configuration>
```

test

http://172.16.0.1:8080/nutch/

3. Lucene

http://lucene.apache.org/

Home | Mirror | Search

Google™ Custom Search Search

4. MG4J

http://mg4j.dsi.unimi.it/

5. PhpDig

http://www.phpdig.net/

PhpDig is a web spider and search engine written in PHP, using a MySQL database and flat file support. PhpDig builds a glossary with words found in indexed pages. On a search query, it displays a result page containing the search keys, ranked by occurrence.

6. Sphinx

http://sphinxsearch.com/

```
sudo apt-get install sphinxsearch
```

/etc/sphinxsearch/sphinx.conf

```
sudo cp /etc/sphinxsearch/sphinx-min.conf.dist /etc/sphinxsearch/sphinx.conf
```

创建测试数据库并导入测试数据

```

$ wget http://sphinxsearch.googlecode.com/svn/trunk/example.sql
$ mysql -h localhost -uroot -p < example.sql
$ mysql -h localhost -uroot -p
CREATE USER 'test'@'localhost' IDENTIFIED BY  '';
GRANT SELECT ON test.* TO 'test'@'localhost';
FLUSH PRIVILEGES;
mysql> quit


$ echo "select * from documents" | mysql -utest -p test
Enter password:
id      group_id        group_id2       date_added      title   content
1       1       5       2011-02-12 15:29:34     test one        this is my test document number
one. also checking search within phrases.
2       1       6       2011-02-12 15:29:34     test two        this is my test document number
two
3       2       7       2011-02-12 15:29:34     another doc     this is another group
4       2       8       2011-02-12 15:29:34     doc number four this is to test groups


```

创建索引

sudo indexer <index>

```

$ sudo indexer test1

Sphinx 0.9.8.1-release (r1533)
Copyright (c) 2001-2008, Andrew Aksyonoff

using config file '/etc/sphinxsearch/sphinx.conf'...
indexing index 'test1'...
collected 4 docs, 0.0 MB
sorted 0.0 Mhits, 100.0% done
total 4 docs, 193 bytes
total 0.012 sec, 16531.05 bytes/sec, 342.61 docs/sec


```

```
$ sudo /etc/init.d/sphinxsearch start
Starting sphinx: Sphinx 0.9.8.1-release (r1533)
Copyright (c) 2001-2008, Andrew Aksyonoff

using config file '/etc/sphinxsearch/sphinx.conf'...
creating server socket on 0.0.0.0:3312
sphinx.
```

测试

search "keyword"

```
$ search test
Sphinx 0.9.8.1-release (r1533)
Copyright (c) 2001-2008, Andrew Aksyonoff

using config file '/etc/sphinxsearch/sphinx.conf'...
index 'test1': query 'test ': returned 3 matches of 3 total in 0.000 sec

displaying matches:
1. document=1, weight=2, group_id=1, date_added=Sat Feb 12 15:29:34 2011
        id=1
        group_id=1
        group_id2=5
        date_added=2011-02-12 15:29:34
        title=test one
        content=this is my test document number one. also checking search within phrases.
2. document=2, weight=2, group_id=1, date_added=Sat Feb 12 15:29:34 2011
        id=2
        group_id=1
        group_id2=6
        date_added=2011-02-12 15:29:34
        title=test two
        content=this is my test document number two
3. document=4, weight=1, group_id=2, date_added=Sat Feb 12 15:29:34 2011
        id=4
        group_id=2
        group_id2=8
        date_added=2011-02-12 15:29:34
        title=doc number four
        content=this is to test groups

words:
1. 'test': 3 documents, 5 hits
```

```
wget http://sphinxsearch.googlecode.com/svn/trunk/api/sphinxapi.php
wget http://sphinxsearch.googlecode.com/svn/trunk/api/test.php
php test.php test
```

7. Mahout

http://mahout.apache.org/

第 9 章 Web Server Optimization

目录

系统配置

1. Intel(R) Xeon(TM) CPU 3.00GHz
2. Memory 4G
3. Ethernet adapter 1000M

1. ulimit

查看 ulimit

```
ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
file size               (blocks, -f) unlimited
pending signals                 (-i) 1024
max locked memory       (kbytes, -l) 32
max memory size         (kbytes, -m) unlimited
```

```
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
stack size              (kbytes, -s) 2048
cpu time               (seconds, -t) unlimited
max user processes              (-u) 77824
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
```

1.1. open files

对于linux系统，所有设备都以映射为设备文件的方式存在，包括硬件（键盘，鼠标，打印机，显示器，串口，并口，USB，硬盘，内存，网卡，声卡，显卡，等等....），还有软件(管道，socket)，访问这些资源，就相当与打开一个文件，

所以"open files"文件数限制很重要，默认值根本不能满足我们。

查看文件打开数

```
$ cat /proc/sys/fs/file-nr

3200    0       197957
已分配文件句柄的数目      已使用文件句柄的数目       文件句柄的最大数目

查看所有进程的文件打开数
    lsof |wc -l
查看某个进程打开的文件数
        lsof -p pid |wc -l
```

临时更改

```
# ulimit -n 65536
or
# ulimit -SHn 65536
or
# echo "65535" > /proc/sys/fs/file-max
```

永久更改

/etc/security/limits.conf

```
nobody          soft    nofile          40960
root            soft    nofile          40960
nobody          hard    nofile          40960
root            hard    nofile          40960
daemon          soft    nofile          40960
daemon          hard    nofile          40960
```

更省事的方法

```
*          soft    nofile          40960
*          hard    nofile          40960
```

最大线程数限制 threads-max

查看当前值

```
# cat /proc/sys/kernel/threads-max
32624
```

设置

有多种方法加大Linux的threads数，下买是临时更改

```
1、sysctl -w kernel.threads-max=65536
2、echo 65536 > /proc/sys/kernel/threads-max
```

永久修改

```
编辑/etc/sysctl.conf
增加
kernel.threads-max = 65536
#sysctl -p 马上生效
```

以上数值仅供参考，随着计算机发展，上面的值已经不太适合，当前流行的服务器。

上一页 下一页
7. Mahout 起始页 2. Memcached

2. Memcached

2.1. 编译安装

http://www.monkey.org/~provos/libevent/

```
cd /usr/local/src/
wget http://www.monkey.org/~provos/libevent-1.4.13-stable.tar.gz
tar zxf libevent-1.4.13-stable.tar.gz
cd libevent-1.4.13-stable
./configure --prefix=/usr/local/libevent-1.4.13-stable
make
make install
make verify

ln -s /usr/local/libevent-1.4.13-stable /usr/local/libevent
ln -s /usr/local/libevent/lib/* /usr/lib/
ln -s /usr/local/libevent/include/* /usr/include/
ln -s /usr/local/libevent/lib/* /usr/local/lib/
ln -s /usr/local/libevent/include/* /usr/local/include/
```

http://www.danga.com/memcached/

```
cd /usr/local/src/
wget http://memcached.googlecode.com/files/memcached-1.4.5.tar.gz
tar zxf memcached-1.4.5.tar.gz
cd memcached-1.4.5
./configure --prefix=/usr/local/memcached-1.4.5 --with-libevent=/usr/local/libevent
make
make install

ln -s /usr/local/memcached-1.4.5/ /usr/local/memcached
ln -s /usr/local/memcached/bin/memcached /usr/sbin/memcached
```

/usr/local/memcached/bin/memcached -d -m 2048 -l 127.0.0.1 -p 11211 -u root -c 15000 -P /tmp/memcached.pid

例 9.1. /etc/init.d/memcached

```

#!/bin/bash
# memcached init file for memcached
#
# chkconfig: - 100 100
# description: a distributed memory object caching system
# author: Neo Chen<openunix@163.com>
#
# processname: /usr/sbin/memcached
# config:
# pidfile: /var/run/memcached

# source function library
. /etc/init.d/functions

OPTIONS="-d -m 2048 -l 127.0.0.1 -p 11211 -u root -c 4096 -P /var/run/memcached"
USER=daemon
RETVAL=0
prog="memcached"

start() {
        echo -n $"Starting $prog: "
        if [ $UID -ne 0 ]; then
                RETVAL=1
                failure
        else
                daemon --user=$USER /usr/sbin/memcached $OPTIONS
                RETVAL=$?
                [ $RETVAL -eq 0 ] && touch /var/lock/subsys/memcached
        fi;
        echo
        return $RETVAL
}

stop() {
        echo -n $"Stopping $prog: "
        if [ $UID -ne 0 ]; then
                RETVAL=1
                failure
        else
                killproc /usr/sbin/memcached
                RETVAL=$?
                [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/memcached
        fi;
        echo
        return $RETVAL
}

reload(){
        echo -n $"Reloading $prog: "
        killproc /usr/sbin/memcached -HUP
        RETVAL=$?
        echo
        return $RETVAL
}

restart(){
        stop
        start
}

condrestart(){
    [ -e /var/lock/subsys/memcached ] && restart
    return 0
}

case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart)
        restart
        ;;
#  reload)
#        reload
#         ;;
  condrestart)
        condrestart
        ;;
  status)
        status memcached
        RETVAL=$?
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        RETVAL=1
esac

exit $RETVAL
```

/etc/init.d/memcached

```
chmod +x /etc/init.d/memcached
```

flush_all指令清空memcache中的数据

```
$ telnet 172.16.3.51 11511
Trying 172.16.3.51...
Connected to 172.16.3.51.
Escape character is '^]'.
flush_all
OK
quit
Connection closed by foreign host.
```

2.2. debian/ubuntu

```
$ sudo apt-get install memcache
```

/etc/memcached.conf

```
$ cat /etc/memcached.conf
# memcached default config file
# 2003 - Jay Bonci <jaybonci@debian.org>
# This configuration file is read by the start-memcached script provided as
# part of the Debian GNU/Linux distribution.

# Run memcached as a daemon. This command is implied, and is not needed for the
# daemon to run. See the README.Debian that comes with this package for more
# information.
-d

# Log memcached's output to /var/log/memcached
logfile /var/log/memcached.log

# Be verbose
# -v

# Be even more verbose (print client commands as well)
# -vv

# Start with a cap of 64 megs of memory. It's reasonable, and the daemon default
# Note that the daemon will grow to this size, but does not start out holding this much
# memory
-m 64

# Default connection port is 11211
-p 11211

# Run the daemon as root. The start-memcached will default to running as root if no
# -u command is present in this config file
-u nobody

# Specify which IP address to listen on. The default is to listen on all IP addresses
# This parameter is one of the only security measures that memcached has, so make sure
# it's listening on a firewalled interface.
-l 127.0.0.1

# Limit the number of simultaneous incoming connections. The daemon default is 1024
# -c 1024

# Lock down all paged memory. Consult with the README and homepage before you do this
# -k

# Return error when memory is exhausted (rather than removing items)
# -M

# Maximize core file limit
# -r
```

restart

```
$ sudo /etc/init.d/memcached restart
```

Home | Mirror | Search

Google™ Custom Search Search

3. khttpd

homepage: http://www.fenrus.demon.nl

4. php.ini

4.1. Resource Limits

Resource Limits

```
;;;;;;;;;;;;;;;;;;;
; Resource Limits ;
;;;;;;;;;;;;;;;;;;;

max_execution_time = 30     ; Maximum execution time of each script, in seconds
max_input_time = 60 ; Maximum amount of time each script may spend parsing request data
;max_input_nesting_level = 64 ; Maximum input variable nesting level
memory_limit = 512M      ; Maximum amount of memory a script may consume (16MB)
```

4.2. File Uploads

```
;;;;;;;;;;;;;;;;
; File Uploads ;
;;;;;;;;;;;;;;;;

; Whether to allow HTTP file uploads.
file_uploads = On

; Temporary directory for HTTP uploaded files (will use system default if not
; specified).
;upload_tmp_dir =

; Maximum allowed size for uploaded files.
upload_max_filesize = 5M
```

4.3. Session Shared

编辑 php.ini 在 [Session]位置添加。

```
extension=memcache.so
memcache.allow_failover = 1
memcache.max_failover_attempts = 20
memcache.chunk_size = 8192
memcache.default_port = 11211

session.save_handler = memcache
session.save_path = "udp://172.16.0.10:11211,tcp://172.16.0.11:11211"
```

4.4. PATHINFO

```
cgi.fix_pathinfo=1
```

5. APC Cache (php-apc - APC (Alternative PHP Cache) module for PHP 5)

```
$ apt-cache search php-apc
php-apc - APC (Alternative PHP Cache) module for PHP 5

$ sudo apt-get install php-apc
```

apc cache 状态监控

http://pecl.php.net/package/APC

下载解包找到apc.php,放到web服务器上

6. Zend Optimizer

http://www.zend.com/

```
tar zxvf ZendOptimizer-3.2.8-linux-glibc21-i386.tar.gz
cd ZendOptimizer-3.2.8-linux-glibc21-i386
./install
```

过程 9.1. 安装 Zend Optimizer

1. 欢迎界面

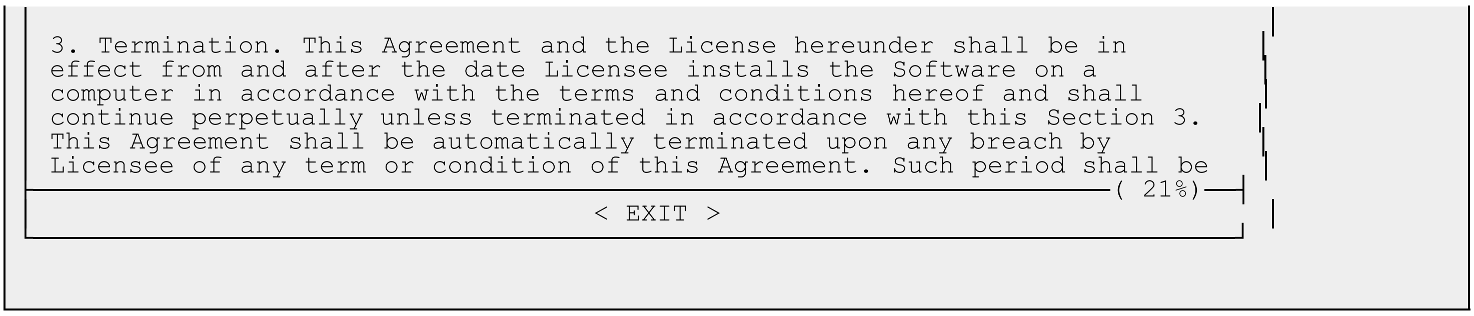
```
 ┌───────────────────── Zend Optimizer 3.2.8 ─────────────────────┐
 │ Welcome to the Zend Optimizer 3.2.8 Installation!               │
 │                                                                 │
 │ For more information regarding this procedure, please see the   │
 │ Zend Optimizer Installation Guide.                              │
 │                                                                 │
 │                                                                 │
 ├─────────────────────────────────────────────────────────────────┤
 │                             <  OK  >                            │
 └─────────────────────────────────────────────────────────────────┘
```

单击 < OK > 按钮

2. LICENSE

Page Down / Page Up 阅读

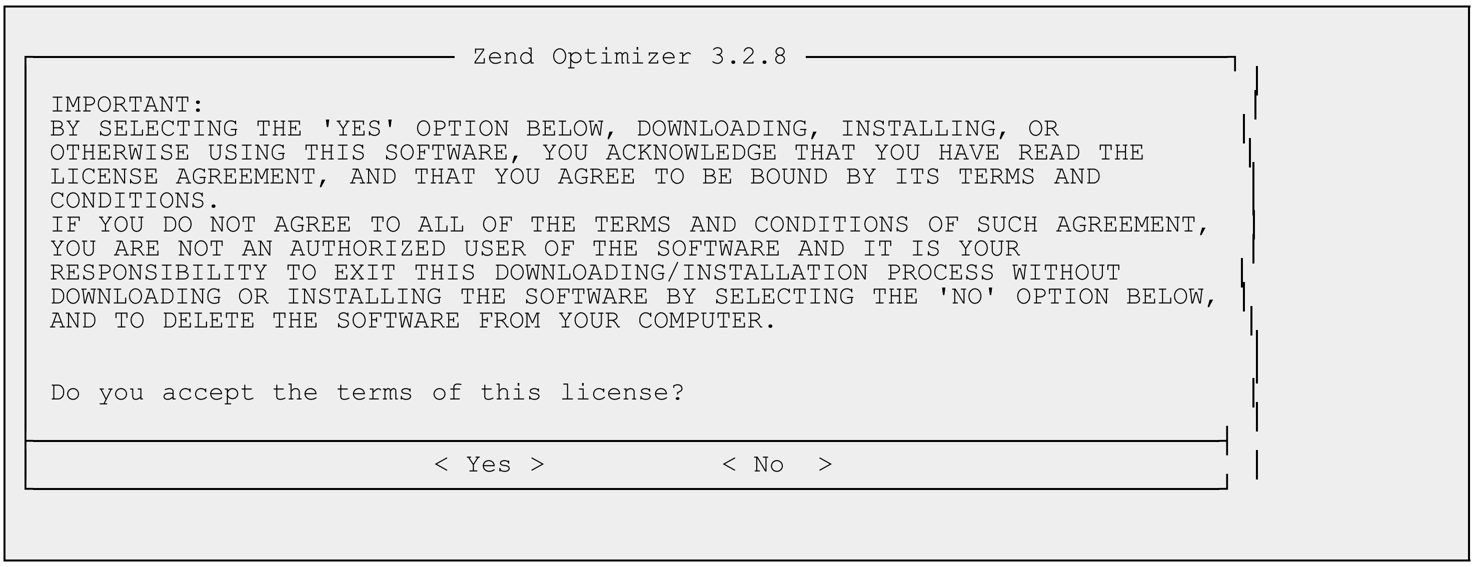
```
 ┌──────────────────────── Zend Optimizer 3.2.8 ─────────────────────────┐
 │ ZEND LICENSE AGREEMENT
 │ Zend Optimizer
 │
 │ ZEND TECHNOLOGIES LTD. ("ZEND") SOFTWARE LICENSE AGREEMENT ("AGREEMENT")
 │
 │ IMPORTANT: READ THESE TERMS CAREFULLY BEFORE INSTALLING THE SOFTWARE KNOWN
 │ AS THE "ZEND OPTIMIZER," AS INSTALLED BY THIS INSTALLATION PROCESS, IN
 │ MACHINE-EXECUTABLE FORM ONLY, AND ANY RELATED DOCUMENTATION (COLLECTIVELY,
 │ THE "SOFTWARE") BY INSTALLING, OR OTHERWISE USING THIS SOFTWARE, YOU (THE
 │ "LICENSEE") ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, AND THAT YOU
 │ AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO ALL
 │ OF THE TERMS AND CONDITIONS OF THIS AGREEMENT, YOU ARE NOT AN AUTHORIZED
 │ USER OF THE SOFTWARE AND IT IS YOUR RESPONSIBILITY TO EXIT THIS
 │ INSTALLATION PROGRAM WITHOUT INSTALLING THE SOFTWARE, OR TO DELETE THE
 │ SOFTWARE FROM YOUR COMPUTER.
 │
 │ 1. License. Subject to the terms and conditions of this Agreement,
 │ including, without limitation, Section 2 hereof, Zend hereby grants to
 │ Licensee, during the Term (as defined below), a limited, a non-exclusive
 │ license (the "License") to: (i) install and operate the Software on a
 │ computer or a computer network owned or operated by Licensee; (ii) make
 │ copies of the Software; and (iii) sublicense and distribute a limited,
 │ non-exclusive sublicense to install, use and sublicense such copies of the
 │ Software, provided that any sub-license granted hereunder shall be subject
 │ to the limitations and restrictions set forth in this Agreement.
 │
 │ 2. Restrictions. Except as otherwise expressly set forth herein, Licensee
 │ or any of its sub-licensees shall not: (a) translate or decompile, or
 │ create or attempt to create, by reverse engineering or otherwise, the
 │ source code form from the object code supplied hereunder; (b) modify,
 │ adapt, translate or create a derivative work from the Software; (c) remove
 │ any proprietary notices, labels, or marks on the Software.
```



```
3. Termination. This Agreement and the License hereunder shall be in
effect from and after the date Licensee installs the Software on a
computer in accordance with the terms and conditions hereof and shall
continue perpetually unless terminated in accordance with this Section 3.
This Agreement shall be automatically terminated upon any breach by
Licensee of any term or condition of this Agreement. Such period shall be
                                                                   ( 21%)
                              < EXIT >
```

单击 < EXIT > 按钮

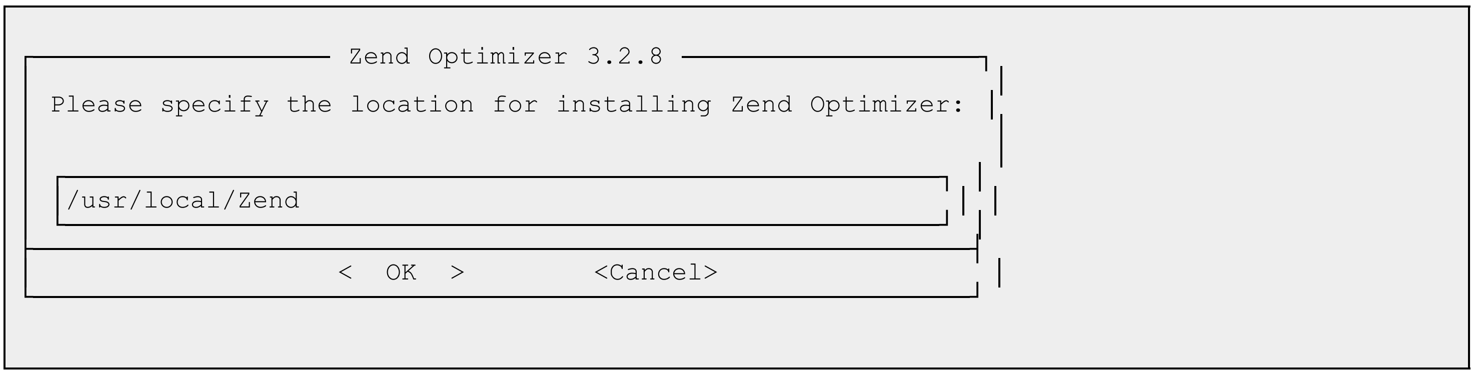
3. 是否接受LICENSE?



```
                     Zend Optimizer 3.2.8
IMPORTANT:
BY SELECTING THE 'YES' OPTION BELOW, DOWNLOADING, INSTALLING, OR
OTHERWISE USING THIS SOFTWARE, YOU ACKNOWLEDGE THAT YOU HAVE READ THE
LICENSE AGREEMENT, AND THAT YOU AGREE TO BE BOUND BY ITS TERMS AND
CONDITIONS.
IF YOU DO NOT AGREE TO ALL OF THE TERMS AND CONDITIONS OF SUCH AGREEMENT,
YOU ARE NOT AN AUTHORIZED USER OF THE SOFTWARE AND IT IS YOUR
RESPONSIBILITY TO EXIT THIS DOWNLOADING/INSTALLATION PROCESS WITHOUT
DOWNLOADING OR INSTALLING THE SOFTWARE BY SELECTING THE 'NO' OPTION BELOW,
AND TO DELETE THE SOFTWARE FROM YOUR COMPUTER.


Do you accept the terms of this license?

                   < Yes >           < No  >
```

单击 < Yes > 按钮

4. Zend Optimizer 安装路径

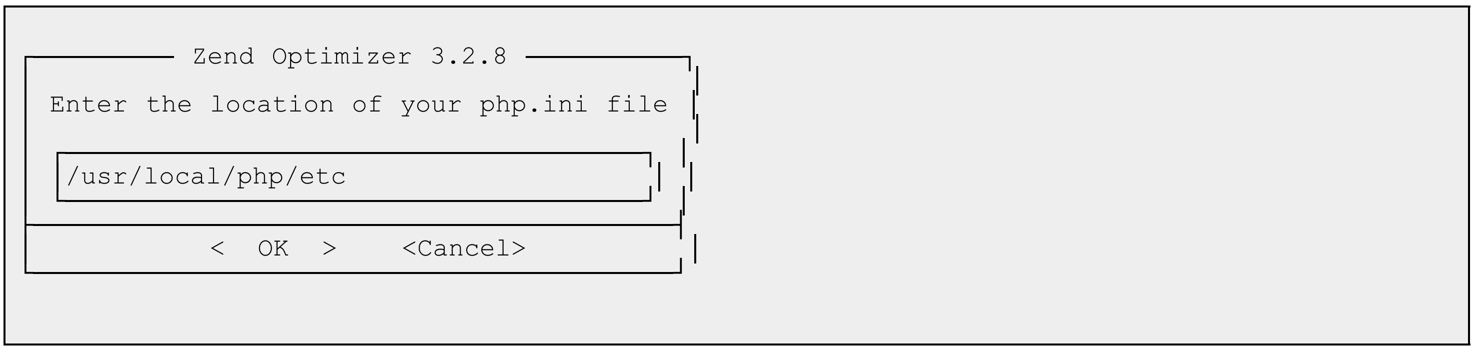


```
               Zend Optimizer 3.2.8
Please specify the location for installing Zend Optimizer:

 /usr/local/Zend

               <  OK  >        <Cancel>
```

单击 < OK > 按钮

建议安装在/usr/local/Zend_3.2.8

5. php.ini 安装路径



```
          Zend Optimizer 3.2.8
Enter the location of your php.ini file

 /usr/local/php/etc

          <  OK  >    <Cancel>
```

输入php.ini安装路径

单击 < OK > 按钮

6. 是否使用了Apache?

```
┌──────── Zend Optimizer 3.2.8 ────────┐
│ Are you using Apache Web server?     │
├──────────────────────────────────────┤
│           < Yes >   < No  >          │
└──────────────────────────────────────┘
```

我的环境是 lighttpd 所以选择 No

单击 < Yes > 按钮

7. 提示信息

```
┌──────────────────────────────────────────────── Zend Optimizer 3.2.8
┌──────────────────────────────────────────────────┐
│
│ The following configuration changes have been made:
│
│
│ - The php.ini file has been relocated from /usr/local/php/etc to
/usr/local/Zend_3.2.8/etc │
│
│ - A symbolic link for the php.ini file has been created in /usr/local/php/etc.
│
│
│ - The original php.ini was backed up to
│  /usr/local/php/etc/php.ini-zend_optimizer.bak
│
│
│
│
├──────────────────────────────────────────────────────────────────────────────────────────────────────────┤
│                                                           <  OK  >
└──────────────────────────────────────────────────────────────────────────────────────────────────────────┘
```

单击 < OK > 按钮

8. 安装完成

```
┌──────────────────────────── Zend Optimizer 3.2.8 ─────────────────────────┐
│ The installation has completed successfully.                              │
│ Zend Optimizer is now ready for use.                                      │
│ You must restart your Web server for the modifications to take effect.   │
├───────────────────────────────────────────────────────────────────────────┤
│                                 <  OK  >                                  │
└───────────────────────────────────────────────────────────────────────────┘
```

单击 < OK > 按钮

7. eaccelerator

```
tar jxvf eaccelerator-0.9.5.3.tar.bz2
cd eaccelerator-0.9.5.3/
/opt/php/bin/phpize
./configure  --enable-eaccelerator=shared --with-php-config=/opt/php/bin/php-config
make
make install
```

第 10 章 varnish - a state-of-the-art, high-performance HTTP accelerator

目录

1. Varnish Install

http://varnish.projects.linpro.no/

1. install

```
$ sudo apt-get install varnish
```

2. /etc/default/varnish

```
$ sudo vim /etc/default/varnish
DAEMON_OPTS="-a :80 \
             -T localhost:6082 \
             -f /etc/varnish/default.vcl \
             -s file,/var/lib/varnish/$INSTANCE/varnish_storage.bin,1G"
```

3. /etc/varnish/default.vcl

```
$ sudo vim /etc/varnish/default.vcl
```

```
backend default {
        .host = "127.0.0.1";
        .port = "8080";
}
```

4. reload

```
$ sudo /etc/init.d/varnish force-reload
 * Stopping HTTP accelerator                            [ OK ]
 * Starting HTTP accelerator
```

上一页 下一页

7. eaccelerator 起始页 2. varnish utility

2. varnish utility

2.1. status

```
$ varnishstat
or
$ varnishstat -n /var/lib/varnish/atom-netkiller/
```

HTTP Head

```
$ curl -I http://bg7nyt.mooo.com/
HTTP/1.1 404 Not Found
X-Powered-By: PHP/5.2.6-3ubuntu4.2
Content-type: text/html
Server: lighttpd/1.4.19
Content-Length: 539
Date: Wed, 23 Sep 2009 00:05:11 GMT
X-Varnish: 938430316
Age: 0
Via: 1.1 varnish
Connection: keep-alive
```

test gzip,defalte

```
$ curl -H Accept-Encoding:gzip,defalte -I http://bg7nyt.mooo.com/
HTTP/1.1 200 OK
X-Powered-By: PHP/5.2.6-3ubuntu4.2
Content-Encoding: gzip
Vary: Accept-Encoding
Content-type: text/html
Server: lighttpd/1.4.19
Date: Wed, 23 Sep 2009 00:08:51 GMT
X-Varnish: 938430335
Age: 0
Via: 1.1 varnish
Connection: keep-alive
```

2.2. varnishadm

help messages

```
$ varnishadm -T 127.0.0.1:6082 help
help [command]
ping [timestamp]
status
start
stop
stats
vcl.load <configname> <filename>
vcl.inline <configname> <quoted_VCLstring>
vcl.use <configname>
vcl.discard <configname>
vcl.list
vcl.show <configname>
param.show [-l] [<param>]
param.set <param> <value>
quit
purge.url <regexp>
purge.hash <regexp>
purge <field> <operator> <arg> [&& <field> <oper> <arg>]...
purge.list
```

2.2.1. 清除缓存

通过Varnish管理端口，使用正则表达式批量清除缓存：

清除所有缓存

```
/usr/local/varnish/bin/varnishadm -T 127.0.0.1:6082 url.purge *$
```

http://bg7nyt.mooo.com/zh-cn/technology/news.html 清除类/zh-cn/下所有缓存

```
/usr/local/varnish/bin/varnishadm -T 127.0.0.1:6082 url.purge /zh-cn/
```

```
/usr/local/varnish/bin/varnishadm -T 127.0.0.1:3500 url.purge w*$
```

2.3. varnishtop

```
varnishtop -i rxurl
varnishtop -i txurl
varnishtop -i RxHeader -I Accept-Encoding
```

2.4. varnishhist

2.5. varnishsizes

Home | Mirror | Search

Google™ Custom Search Search

3. log file

log file

```
$ sudo vim  /etc/default/varnishlog
VARNISHLOG_ENABLED=1
$ sudo /etc/init.d/varnishlog start
 * Starting HTTP accelerator log deamon    [ OK ]

$ sudo vim  /etc/default/varnishncsa
VARNISHNCSA_ENABLED=1
$ sudo /etc/init.d/varnishncsa start
 * Starting HTTP accelerator log deamon    [ OK ]
```

上一页 上一级 下一页

2. varnish utility 起始页 4. Varnish Configuration Language - VCL

4. Varnish Configuration Language - VCL

Varnish配置文件VCL中的函数详解

内置的例程

```
vcl_recv
有请求到达后成功接收并分析时被调用，一般以以下几个关键字结束。
error code [reason] 返回code给客户端，并放弃处理该请求
pass 进入pass模式，把控制权交给vcl_pass
pipe 进入pipe模式，把控制权交给vcl_pipe
lookup 在缓存里查找被请求的对象，根据查找结果把控制权交给vcl_hit或vcl_miss

vcl_pipe
进入pipe模式时被调用。请求被直接发送到backend，后端和客户端之间的后继数据不进行处理，只是简单传递，直到一方关闭连接。一般以以下几个关键字结束。
error code [reason]
pipe

vcl_pass
进入pass模式时被调用。请求被送到后端，后端应答数据送给客户端，但不进入缓存。同一连接的后继请求正常处理。一般以以下几个关键字结束。
error code [reason]
pass

vcl_hash
目前不使用

vcl_hit
在lookup以后如果在cache中找到请求的内容事调用。一般以以下几个关键字结束。
error code [reason]
pass
deliver 将找到的内容发送给客户端，把控制权交给vcl_deliver.

vcl_miss
lookup后但没有找到缓存内容时调用，可以用于判断是否需要从后端服务器取内容。一般以以下几个关键字结束。
error code [reason]
pass
fetch 从后端取得请求的内容，把控制权交给vcl_fetch.


vcl_fetch
从后端取得内容后调用。一般以以下几个关键字结束。
error code [reason]
pass
insert 将取到的内容插入缓存，然后发送给客户端，把控制权交给vcl_deliver


vcl_deliver
缓存内容发动给客户端前调用。一般以以下几个关键字结束。
error code [reason]
deliver 内容发送给客户端

vcl_timeout
在缓存内容到期前调用。一般以以下几个关键字结束。
fetch 从后端取得该内容
discard 丢弃该内容



vcl_discard
由于到期或者空间不足而丢弃缓存内容时调用。一般以以下几个关键字结束。
discard 丢弃
keep 继续保留在缓存里

如果这些内置例程没有被定义，则执行缺省动作


一些内置的变量
now 当前时间，标准时间点（1970？）到现在的秒数

backend.host 后端的IP或主机名
backend.port 后端的服务名或端口

请求到达后有效的变量
client.ip 客户端IP
```

```
server.ip 服务端IP
req.request 请求类型，比如GET或者HEAD或者POST
req.url 请求的URL
req.proto 请求的HTTP版本号
req.backend 请求对应的后端
req.http.header 对应的HTTP头

往后段的请求时有效的变量
bereq.request 比如GET或HEAD
bereq.url URL
bereq.proto 协议版本
bereq.http.header HTTP头

从cache或后端取到内容后有效的变量
obj.proto HTTP协议版本
obj.status HTTP状态代码
obj.response HTTP状态信息
obj.valid 是否有效的HTTP应答
obj.cacheable 是否可以缓存的内容，也就是说如果HTTP返回是200、203、300、301、302、404、410并且有非0的生存
期，则为可缓存
obj.ttl 生存期，秒
obj.lastuse 上一次请求到现在间隔秒数

对客户端应答时有效的变量
resp.proto response的HTTP版本
resp.status 回给客户端的HTTP状态代码
resp.response 回给客户端的HTTP状态信息
resp.http.header HTTP头
```

5. example

例 10.1. default.vcl

```
neo@netkiller:/etc/varnish$ cat default.vcl
# This is a basic VCL configuration file for varnish.  See the vcl(7)
# man page for details on VCL syntax and semantics.
#
# Default backend definition.  Set this to point to your content
# server.
#
backend default {
    .host = "127.0.0.1";
    .port = "8080";
}
#
# Below is a commented-out copy of the default VCL logic.  If you
# redefine any of these subroutines, the built-in logic will be
# appended to your code.
#
sub vcl_recv {
    if (req.http.x-forwarded-for) {
        set req.http.X-Forwarded-For =
            req.http.X-Forwarded-For ", " client.ip;
    } else {
        set req.http.X-Forwarded-For = client.ip;
    }
    if (req.request != "GET" &&
      req.request != "HEAD" &&
      req.request != "PUT" &&
      req.request != "POST" &&
      req.request != "TRACE" &&
      req.request != "OPTIONS" &&
      req.request != "DELETE") {
        /* Non-RFC2616 or CONNECT which is weird. */
        return (pipe);
    }
    if (req.request != "GET" && req.request != "HEAD") {
        /* We only deal with GET and HEAD by default */
        return (pass);
    }
    if (req.http.Authorization || req.http.Cookie) {
        /* Not cacheable by default */
/*        return (pass);*/
        return (lookup);
    }
    return (lookup);
}

sub vcl_pipe {
    # Note that only the first request to the backend will have
    # X-Forwarded-For set.  If you use X-Forwarded-For and want to
    # have it set for all requests, make sure to have:
    # set req.http.connection = "close";
    # here.  It is not set by default as it might break some broken web
    # applications, like IIS with NTLM authentication.
    return (pipe);
}

sub vcl_pass {
    return (pass);
}

sub vcl_hash {
    set req.hash += req.url;
    if (req.http.host) {
        set req.hash += req.http.host;
    } else {
        set req.hash += server.ip;
    }
    return (hash);
}

sub vcl_hit {
    if (!obj.cacheable) {
        return (pass);
    }
    return (deliver);
}
```

```
sub vcl_miss {
    return (fetch);
}

sub vcl_fetch {
    if (!beresp.cacheable) {
        return (pass);
    }
    if (beresp.http.Set-Cookie) {
#         return (pass);
        return (deliver);
    }
    return (deliver);
}

sub vcl_deliver {
    return (deliver);
}
#
# sub vcl_error {
#     set obj.http.Content-Type = "text/html; charset=utf-8";
#     synthetic {"
# <?xml version="1.0" encoding="utf-8"?>
# <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
#  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
# <html>
#   <head>
#     <title>"} obj.status " " obj.response {"</title>
#   </head>
#   <body>
#     <h1>Error "} obj.status " " obj.response {"</h1>
#     <p>"} obj.response {"</p>
#     <h3>Guru Meditation:</h3>
#     <p>XID: "} req.xid {"</p>
#     <hr>
#     <p>Varnish cache server</p>
#   </body>
# </html>
# "};
#     return (deliver);
# }
```

第 11 章 Traffic Server

目录

1. Install

```
yum install gcc gcc-c++ make autoconf -y
yum -y install  tcl lzma tcl-devel expat expat-devel pcre-devel perl perl-devel
```

```
cd /usr/local/src/
wget http://mirror.bjtu.edu.cn/apache//trafficserver/trafficserver-3.0.1.tar.bz2
tar -xvjf trafficserver-3.0.1.tar.bz2
```

```

cd trafficserver-3.0.1
./configure --prefix=/srv/trafficserver-3.0.1 && make && make install

```

2.

```
修改配置
vi records.config
  CONFIG proxy.config.proxy_name STRING cache1                  ### 修改成cache的server name即可
  CONFIG proxy.config.cluster.ethernet_interface STRING eth0    ### 修改成需要侦听的interface名称，
默认是 null
  CONFIG proxy.config.admin.user_id STRING nobody               ### 用来运行 traffic server 的用
户,默认是nobody
  CONFIG proxy.config.http.server_port INT 80                   ### traffic server 侦听的端口，默
认是8080

vi cache.config
dest_domain=www.xiu.com scheme=http     revalidate=2h

vi remap.conf
map http://www.xiu.com  http://10.0.0.51  #前一个是用户访问的地址，后一个是源站点的IP，或者域名

配置变更应用生效
/srv/ts/bin/traffic_line -x

启动服务

/srv/ts/bin/trafficserver start


./traffic_shell
show
show:cache
show:cache-stats
show:proxy-stats

./logstats -i www.xiu.com

如果服务器down掉，默认会生成core文件，在/ts
使用

ts/bin/traffic_server -c core.1234
```

第 12 章 Cherokee

目录

1. Installing Cherokee

```
apt-get install cherokee
```

Cherokee can be configured through a web-based control panel which we can start as follows:

```
cherokee-admin -b
```

cherokee script

```
/etc/init.d/cherokee restart
```

Home | Mirror | Search

Google™ Custom Search Search

第13章 Jetty

上一页 下一页

第12章 Cherokee 起始页 第14章 Other Web Server

上一页

Home | Mirror | Search

Google™ Custom Search Search

第 14 章 Other Web Server

目录

1. Python SimpleHTTPServer

```
python -m SimpleHTTPServer &
```

```
curl http://localhost:8000/
```

上一页

第 13 章 Jetty 起始页