

# LEMP 构建高性能 WEB 服务器（第三版）

作者:NetSeek <http://www.linuxtone.org> (IT运维专家网|集群架构|性能调优)

欢迎转载,转载时请务必以超链接形式标明文章原始出处和作者信息及本声明.

更新时间:2009-01-5

## 平台搭建环境:

CentOS5.2 32/x86\_64 GNU/Linux (32/64 操作系统均通过, 推荐使用 64 位操作系统)

目标: 构建高性能的 LEMP 环境提供 WEB 服务

## 一、系统安装

### 1. 系统分区建议

<code>/boot</code>	100M	(大约 100 左右)
<code>SWAP</code>		物理内存的 2 倍(如果你的物理内存大于 4G, 分配 4G 即可)
<code>/</code>		分区 15~20G
<code>/usr/local</code>	20G	(用于安装软件)
<code>/data</code>		剩余所有空间.

【注】具体分区请根据相关业务划分, 具体安装本文不作介绍安装过程中选择最少的包, 采用文本模式安装请不要安装图形。

### 2. 系统初始化

系统初始化脚本(根据具体需求关闭不需要的服务)

```
# sh init_network.sh //下载本文提供的 scripts.tar.gz 包, 即可找到此脚本.
```

## 二、编译安装基本环境

### 1. 安装准备

#### 1.1 系统软件包安装规范

##### 系统约定

软件源代码包存放位置	<code>/usr/local/src</code>
源码包编译安装位置(prefix)	<code>/usr/local/software_name</code>
脚本以及维护程序存放位置	<code>/usr/local/sbin</code>
MySQL 数据库位置	<code>/data/mysql/data</code> (可按情况设置)
网站根目录	<code>/data/www/wwwroot</code> (可按情况设置)
虚拟主机日志根目录	<code>/data/logs</code> (可按情况设置)
Nginx 运行账户	<code>www:www</code>
<code>install_software_name.sh</code>	//存放编译参数脚本习惯将所有编译脚本存放在 <code>install_software_name.sh</code> 便于升级和更新软件.

##### 创建网站账号及相关存放目录

```
#add_webuser.sh  
/usr/sbin/groupadd www
```

```
/usr/sbin/useradd -g www www
mkdir -p /data/www/wwwroot
chmod +w /data/www/wwwroot
chown www:www /data/www/wwwroot -R
```

## 1.2 系统环境部署及调整

### 检查系统是否正常

```
# tail -n100 /var/log/messages      (检查有无系统级错误信息)
# dmesg      (检查硬件设备是否有错误信息)
# ifconfig  (检查网卡设置是否正确)
# ping www.linuxtone.org      (检查网络是否正常)
```

## 1.3 使用 yum 程序安装所需开发包 (以下为标准的 RPM 包名称)

添加国内镜像源加速软件安装下载速度请参照:<http://bbs.linuxtone.org/thread-158-1-1.html>

```
yum -y install ntp vim-enhanced gcc gcc-c++ gcc-g77 flex bison autoconf automake
bzip2-devel \
ncurses-devel openssl-devel libtool* zlib-devel libxml2-devel libjpeg-devel libpng-devel
libtiff-devel \
fontconfig-devel freetype-devel libXpm-devel gettext-devel curl curl-devel pam-devel \
e2fsprogs-devel krb5-devel libidn libidn-devel
```

## 1.4 定时校正服务器时钟, 定时与中国国家授时中心授时服务器同步

```
# crontab -e
```

加入一行:

```
15 3 * * * /usr/sbin/ntpdate 210.72.145.44 > /dev/null 2>&1
```

## 1.5 下载编译相关的源码包.

#vi list 在 list 文件里填入以后下载地址列表.

官方下载软件包名:

```
#nginx
ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-7.8.tar.gz
http://sysoev.ru/nginx/nginx-0.7.30.tar.gz
#PHP-GD-LIB
http://www.libgd.org/releases/gd-2.0.35.tar.bz2
http://ftp.gnu.org/pub/gnu/libiconv/libiconv-1.12.tar.gz
http://jaist.dl.sourceforge.net/sourceforge/mcrypt/libmccrypt-2.5.8.tar.bz2
http://downloads.sourceforge.net/mhash/mhash-0.9.9.9.tar.gz?modtime=1228695303&big\_mirror=0
http://jaist.dl.sourceforge.net/sourceforge/mcrypt/mcrypt-2.6.7.tar.gz
#MySQL
http://dev.mysql.com/get/Downloads/MySQL-5.1/mysql-5.1.30.tar.gz/from/http://mysql.cdpa.nsysu.edu.tw/
#php
http://cn.php.net/get/php-5.2.8.tar.bz2/from/cn2.php.net/mirror
```

```
http://php-fpm.anight.org/downloads/head/php-5.2.8-fpm-0.5.10.diff.gz  
#PHP-EXT  
http://pecl.php.net/get/memcache-2.2.4.tgz  
http://pecl.php.net/get/PDO\_MYSQL-1.0.2.tgz  
http://bart.eaccelerator.net/source/0.9.5.3/eaccelerator-0.9.5.3.tar.bz2
```

```
#wget -i list
```

## 2. 编译安装软件包

源码编译安装所需包(Source)

### 1.1 编译安装 Nginx

```
tar zxvf pcre-7.8.tar.gz  
cd pcre-7.8  
./configure && make && make install
```

```
tar zxvf nginx-0.7.30.tar.gz  
cd nginx-0.7.30  
./configure --user=www --group=www --prefix=/usr/local/nginx/  
--with-http_stub_status_module --with-openssl=/usr/lib  
make && make install
```

### 1.2 编译安装 GD 及相关的 LIB.

```
tar jxvf gd-2.0.35.tar.bz2  
cd gd-2.0.35  
./configure --prefix=/usr/local/gd2 && make && make install
```

```
tar zxvf libiconv-1.12.tar.gz  
cd libiconv-1.12  
./configure --prefix=/usr && make && make install
```

```
tar jxvf libmcrypt-2.5.8.tar.bz2  
cd libmcrypt-2.5.8  
./configure --prefix=/usr && make && make install
```

```
tar zxvf mhash-0.9.9.tar.gz  
cd mhash-0.9.9  
./configure --prefix=/usr && make && make install
```

```
#vi /etc/ld.so.conf 添加/usr/lib  
include ld.so.conf.d/*.conf  
/usr/lib
```

### # ldconfig

【注】ldconfig 命令的用途,主要是在默认搜寻目录(/lib 和/usr/lib)以及动态库配置文件/etc/ld.so.conf 内所列的目录,而当用户安装了一个新的动态链接库时,就需要手工运行这个命令。

```
tar zxvf mcrypt-2.6.7.tar.gz
cd mcrypt-2.6.7
./configure --prefix=/usr && make && make install
```

## 1.3 编译安装 MySQL

#sh install\_mysql.sh //具体脚本如下:

```
CHOST="x86_64-pc-linux-gnu"
CFLAGS="-march=nocona -O3 -pipe"
CXXFLAGS="${CFLAGS}"
./configure \
    "--prefix=/usr/local/mysql" \
    "--localstatedir=/data/mysql/data" \
    "--with-comment=Source" \
    "--with-server-suffix=-Linuxtone.Org" \
    "--with-mysqld-user=mysql" \
    "--without-debug" \
    "--with-big-tables" \
    "--with-charset=gbk" \
    "--with-collation=gbk_chinese_ci" \
    "--with-extra-charsets=all" \
    "--with-pthread" \
    "--enable-static" \
    "--enable-thread-safe-client" \
    "--with-client-ldflags=-all-static" \
    "--with-mysqld-ldflags=-all-static" \
    "--enable-asm" \
    "--with-plugins=all" \
    "--without-ndb-debug"

make
make install
useradd mysql -d /data/mysql -s /sbin/nologin
/usr/local/mysql/bin/mysql_install_db --user=mysql
cd /usr/local/mysql
chown -R root:mysql .
mkdir -p /data/mysql/data
chown -R mysql /data/mysql/data
cp share/mysql/my-huge.cnf /etc/my.cnf
cp share/mysql/mysql.server /etc/rc.d/init.d/mysqld
chmod 755 /etc/rc.d/init.d/mysqld
```

```
chkconfig --add mysqld
/etc/rc.d/init.d/mysqld start
```

```
cd /usr/local/mysql/bin
for i in *; do ln -s /usr/local/mysql/bin/$i /usr/bin/$i; done
```

【注】编译 mysql 前请正确配置好主机名，否则不能正常启动。

例如：

```
[root@lemp mysql-5.1.30]# cat /etc/hosts
```

```
# Do not remove the following line, or various programs
```

```
# that require network functionality will fail.
```

```
127.0.0.1    localhost.localdomain localhost
```

```
192.168.1.104      lemp.linuxtone.org
```

【注】将本文提供下载的 scripts.tar.gz 包中的 my.cnf 替换你的/etc/my.cnf,并根据你的硬件参数做相关调整,并可以利用包中的 tuning-primer.sh 做初步性能调优。

#### 1.4 编译安装 PHP-FastCGI

```
tar jxvf php-5.2.8.tar.bz2
gzip -cd php-5.2.8-fpm-0.5.10.diff.gz | patch -d php-5.2.8 -p1
cd php-5.2.8
```

#sh install\_php.sh //具体脚本如下

```
CHOST="i686-pc-linux-gnu"
CFLAGS="-march=prescott -O3 -pipe -fomit-frame-pointer"
CXXFLAGS="${CFLAGS}"
./configure \
    "--prefix=/usr/local/php-fcgi" \
    "--enable-fastcgi" \
    "--enable-fpm" \
    "--enable-discard-path" \
    "--enable-force-cgi-redirect" \
    "--with-config-file-path=/usr/local/php-fcgi/etc" \
    "--enable-zend-multibyte" \
    "--with-mysql=/usr/local/mysql" \
    "--with-libxml-dir" \
    "--with-xmlrpc" \
    "--with-gd=/usr/local/gd2" \
    "--with-jpeg-dir" \
    "--with-png-dir" \
    "--with-bz2" \
    "--with-freetype-dir" \
    "--with-iconv-dir" \
    "--with-zlib-dir" \
    "--with-curl" \
    "--with-curlwrappers" \
```

```
--with-openssl" \  
--with-mcrypt" \  
--with-mhash" \  
--enable-pcntl" \  
--enable-sockets" \  
--enable-sysvsem" \  
--enable-inline-optimization" \  
--enable-soap" \  
--enable-gd-native-ttf" \  
--enable-ftp" \  
--enable-mbstring" \  
--enable-exif" \  
--disable-debug" \  
--disable-ipv6"  
  
#sed -i 's#-lz -lm -lxml2 -lz -lm -lxml2 -lz -lm -lcrypt#& -liconv#' Makefile  
make ZEND_EXTRA_LIBS='-liconv'  
make install  
cp php.ini-dist /usr/local/php-fcgi/etc/php.ini
```

### 1.5 编译安装 PHP 扩展

```
tar zxvf memcache-2.2.4.tgz  
cd memcache-2.2.4  
/usr/local/php-fcgi/bin/phpize  
./configure --with-php-config=/usr/local/php-fcgi/bin/php-config  
make && make install
```

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

```
tar zxvf PDO_MYSQL-1.0.2.tgz  
cd PDO_MYSQL-1.0.2  
/usr/local/php-fcgi/bin/phpize  
./configure --with-php-config=/usr/local/php-fcgi/bin/php-config  
--with-pdo-mysql=/usr/local/mysql  
make  
make install
```

## 1.6 配置 php.ini

```
#sh init_php.sh //执行 PHP 配置脚本
```

【注】下载包中提供了此脚本,主要是添加自动添加 PHP 扩展及 PHP 安全相关设置,用户可以编辑此文件根据具体情况做调整。

```
#vi /usr/local/php-fcgi/etc/php.ini //在最后加上
```

```
[eAccelerator]
extension="eaccelerator.so"
eaccelerator.shm_size="32"
eaccelerator.cache_dir="/cache_tmpfs/eaccelerator"
eaccelerator.enable="1"
eaccelerator.optimizer="1"
eaccelerator.check_mtime="1"
eaccelerator.debug="0"
eaccelerator.filter=""
eaccelerator.shm_max="0"
eaccelerator.shm_ttl="0"
eaccelerator.shm_prune_period="0"
eaccelerator.shm_only="0"
eaccelerator.compress="1"
eaccelerator.compress_level="9"
```

## 1.7 安装 ZendOptimizer-3.3.3

```
tar zxvf ZendOptimizer-3.3.3-linux-glibc23-x86_64.tar.gz
cd ZendOptimizer-3.3.3-linux-glibc23-x86_64
./install
```

解压后进入目录./install, 安提示选择相关的目录及配置文件存放目录即可。

## 1.8 配置 php-fpm.conf

```
rm -rf /usr/local/php-fcgi/etc/php-fpm.conf
```

将本文提供的 php-fpm.conf 文件上传此目录。

```
/usr/local/php-fcgi/sbin/php-fpm start //启动服务查看是否报错,不报错继续。
```

## 1.9 配置 nginx/conf

```
#cd /usr/local/nginx
#mv conf conf_old
#mkdir -p conf/{vhost,conf_example}
```

#将本文提供的 conf 目录替换成当前 conf 目录.在下载的 scripts.tar.gz 包里.

目录结构介绍:

```
/usr/local/nginx/conf //存放主配置文件
```

```
/usr/local/nginx/conf/vhost //存放虚拟主机配置文件.
```

```
/usr/local/nginx/conf/conf_example //存放一些经典的配置示例{虚拟主机, 防盗链, 访问控制,重定向, 负载均衡等}
```

```
[root@lemp mysql]# /usr/local/nginx/sbin/nginx -t //验证是否有配置语法错误.
2009/01/06 09:36:12 [info] 23353#0: the configuration file /usr/local/nginx/conf/nginx.conf
syntax is ok
2009/01/06 09:36:12 [info] 23353#0: the configuration file /usr/local/nginx/conf/nginx.conf
was tested successfully
```

```
#chmod a+x /etc/init.d/nginx
```

//将文本提供的 nginx 脚本上传至/etc/init.d/目录下

【注】此脚本已经含有 php-fpm 的管理.直接利用此脚本即可管理所有服务.

```
# /etc/init.d/nginx stop //停止 nginx 服务
```

```
# /etc/init.d/nginx start //启动 nginx 服务
```

## 2.0 验证服务环境

配置完成虚拟主机以后, 编辑 php.ini 把 disable\_functions 里的 phpinfo 去掉, 仍后重启服务, 利用 Phpinfo 函数查看环环配置。

编写 phpinfo.php

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

<http://ip/phpinfo.php>

即可查看当前配置环境. 查看完成后出于安全考虑请再将此函数禁掉.

## 三、服务器简单优化

### 1.简单优化磁盘 I/O

```
# vi /etc/fstab //加入 notaime
```

LABEL=/data	/data	ext3	defaults,noatime	0 2
LABEL=/boot1	/boot	ext3	defaults,noatime	1 2

```
#mount -oremount /data //重新挂载/data 分区.
```

### 2.简单优化 Linux 内核参数 (更多的请根据服务器状况做相关调整)

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

```
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_keepalive_time = 300
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 1
net.ipv4.ip_local_port_range = 5000 65000
```

```
sysctl -p
```

### 3.其它优化根据具体情况还有硬件情况做调整.

更多的优化请参考: <http://www.linuxtone.org/forum-31-1.html>或联系我们.

#### 四、相关参考及介绍.

1. **Nginx Tips(Nginx应用技巧含优化)** <http://bbs.linuxtone.org/thread-1241-1-1.html>
2. **LEMP构建高性能WEB服务器（第二版）** <http://bbs.linuxtone.org/thread-123-1-1.html>
3. **如果你想构建可用集群方案请参考** <http://www.linuxtone.org/thread-1402-1-1.html>
4. **如果你想学习服务器监控请参考** <http://www.linuxtone.org/forum-20-1.html>

#### 五、相关软件包及脚本下载

<http://www.linuxtone.org/lemp/lemp.tar.gz> 所有软件包如果发现不能下载请去官方下载。  
<http://www.linuxtone.org/lemp/scripts.tar.gz> 文中所有脚本及配置范例。



如果你在建站配置服务过程中遇到什么问题?  
欢迎你与我们联系或到我们的网站讨论!  
同时希望更多的 Linux 爱好者能加入我们, 和我们一起建设这个技术交流平台!

IT运维专家网 <http://www.linuxtone.org> 自由+平等! 互助+友爱!

Email: [netseek@linuxtone.org](mailto:netseek@linuxtone.org)

#### 六、常见错误（疑问）解答 [守住每一天] 补充:

- 1、eaccelerator 与 xcache 差在那?  
当缓存的 PHP 二进制文件太多, 以至于超过 shm\_size 后, eAccelerator 在性能方面要高于 Xcache
- 2、按文档安装完成了结果出 502 Bad Gateway ?  
请查看<http://www.linuxtone.org> 相关版块
- 3、nginx status 解释  
active connections -- 对后端发起的活动连接数  
server accepts handled requests -- nginx 总共处理了 43629083 个连接, 成功创建 43629083 次握手 (证明中间没有失败的), 总共处理了 259552136 个请求  
reading -- nginx 读取到客户端的 Header 信息数  
writing -- nginx 返回给客户端的 Header 信息数  
waiting -- 开启 keep-alive 的情况下, 这个值等于 active - (reading + writing), 意思就是 Nginx 已经处理完成, 正在等候下一次请求指令的驻留连接。在访问效率高, 请求很快被处理完毕的情况下, Waiting 数比较多是正常的。如果 reading + writing 数较多, 则说明并发访问量非常大, 正在处理过程中。
- 4、EA 或 xcache 和 zend 能一起使用?  
可以
- 5、按脚本执行 `sed -i -e 's/^LANG=.* /LANG="en"/' /etc/sysconfig/i18n` 后。可能会遇到如下错误:  
perl: warning: Please check that your locale settings:  
LANGUAGE = (unset),

```
LC_ALL = (unset),
```

```
LANG = "en_US.UTF-8:en_US:en_US.ISO-8859-1"
```

are supported and installed on your system.

perl: warning: Falling back to the standard locale ("C").

解决办法:

```
LC_ALL=C;export LC_ALL ; LC_CTYPE=C ; export LC_CTYPE;
```