安装请浏览全文再进行安装．将需要安装的文件通过wget从网络下载．

更新基础库：

yum -y install openssl\*

yum -y install gcc gcc-c++ autoconf libjpeg libjpeg-devel libpng libpng-devel freetype freetype-devel libxml2 libxml2-devel zlib zlib-devel glibc glibc-devel glib2 glib2-devel bzip2 bzip2-devel ncurses ncurses-devel curl curl-devel e2fsprogs e2fsprogs-devel krb5 krb5-devel libidn libidn-devel openssl openssl-devel openldap openldap-devel nss\_ldap openldap-clients openldap-servers make

yum -y install gd gd2 gd-devel gd2-devel

## MySQL

参考压缩包的INSTALL文件，逐步安装即可．

开机服务自动启动：

cp support-files/mysql.server /etc/init.d/mysqld

chkconfig –level 235 mysqld on

设计 chkconfig 的目的就是用来管理系统初始化的时候启动的服务。还能通过管理 init 脚本的符号连接得以最终控制启动关闭时的系统任务。

## Nginx安装：

作为 Web 服务器：相比Apache，Nginx 使用更少的资源，支持更多的并发连接，体现更高的效率，这点使 Nginx 尤其受到虚拟主机提供商的欢迎。能够支持高达 50,000 个并发连接数的响应，感谢 Nginx 为我们选择了 epoll and kqueue 作为开发模型.

Nginx会用到PCRE(正则表达式，Perl Compatible Regular Expressions)，这个是必须提前安装，必须：

./configure --enable-utf8 --enable-unicode-properties

openssl也需要安装，在HTTPSSL也是会用到，默认configure就行

需要做动态库引用，它的默认安装路径为/usr/loca/ssl

Nginx安装时指定一下openssl的路径，同时可以把nginx的HTTPSSL和状态监控模块也安装上去．详细的安装选项可以通过./configure –help查询．

./configure --prefix=/home/apps/nginx-1.4.6 --with-http\_stub\_status\_module --with-http\_ssl\_module --with-http\_realip\_module --with-http\_image\_filter\_module

监控模块需要配置Nginx.conf：

*location /nginx\_status {  
stub\_status on;  
access\_log off;  
allow 192.168.1.1;#设置为可访问该状态信息的ip  
deny all;  
}*

配置成功后提示：

*Configuration summary*

*+ using system PCRE library*

*+ using OpenSSL library: /usr/local/ssl*

*+ md5: using OpenSSL library*

*+ sha1: using OpenSSL library*

*+ using system zlib library*

*nginx path prefix: "/usr/local/nginx"*

*…..*

**nginx空主机头配置：**

server {

listen 80 default;

return 500;

}

## PHP安装

yum -y install gcc gcc-c++ autoconf libjpeg libjpeg-devel libpng libpng-devel freetype freetype-devel libxml2 libxml2-devel zlib zlib-devel glibc glibc-devel glib2 glib2-devel bzip2 bzip2-devel ncurses ncurses-devel curl curl-devel

<可选>mhash是php hash加密算法很强大的扩展，但是php5.3.0以后就不包含这个程序包了，需要自己去下载，<http://sourceforge.net/projects/mhash>。

需要做动态库引用，它的默认安装路径为/usr/loca/mhash

<可选>libmcrypt/mcrypt：libmcrypt是加密算法扩展库。支持DES, 3DES, RIJNDAEL, Twofish, IDEA, GOST, CAST-256, ARCFOUR, SERPENT, SAFER+等算法

注意：安装libmcrypt时：

cd libltdl/

./configure --enable-ltdl-install

Libxml与zlib有冲突，在安装libxml之前最好把zlib都删除一下，在６４位系统下面一定要这样操作．

rpm -e zlib-1.2.3-4.el5 --nodeps --allmatches

zlib:使用DEFLATE算法，最初是为libpng函式库所写的，后来普遍为许多软件所使用。7-zip，libpng，apache/nginx，openssh/openssl，svn/git/cvs，rpm等都在zlib的deflate算法．

目前zlib仅支持一个LZ77的变种算法，DEFLATE的算法。

gzip和deflate

<http://apps.hi.baidu.com/share/detail/30787631>

其他压缩：bzip2是一个基于Burrows-Wheeler 变换的无损压缩软件，压缩效果比传统的LZ77/LZ78压缩演算法来得好。它免费提供，具有高质量的数据压缩能力。tar xjvf 解压

Libiconv：libiconv库为需要做转换的应用提供了一个iconv()的函数，以实现一个字符编码到另一个字符编码的转换。

图片格式支持libPNG,JPG库

字体库：freetype, fontconfig

FreeType库：是一个完全免费(开源)的、高质量的且可移植的字体引擎，它提供统一的接口来访问多种字体格式文件，包括TrueType, OpenType, Type1, CID, CFF, Windows FON/FNT, X11 PCF等

Fontconfig:

GD: gd库是php处理图形的扩展库，gd库提供了一系列用来处理图片的API，使用GD库可以处理图片，或者生成图片。 在网站上GD库通常用来生成缩略图或者用来对图片加水印或者对网站数据生成报表。

### PHP安装：

在5.2.17时需下载PHP及PHP-FPM，而在PHP5.3.3之后直接将php-fpm加入至内核．不用再进行补丁．

PHP-FPM( (FastCGI Process Manager) )是一个PHP FastCGI管理器，用来控制PHP-CGI的FastCGI进程，它是PHP源代码的一个补丁，

在PHP5.2.17版本下面，它是以static方式运作，进程是固定，在PHP5.3之后，采用了动态的管理，空闲时自动关闭进程到配置的最小数目，以此合理来分配系统资源．通常一个PHP-FPM进程通常会占用8-16M的内存(张宴博客中提到的是20M/FPM)．如果进程数目控制为300，大概会占用4G内存左右．其他内存腾出给到数据库或其他业务进程．

备选：spawn-fcgi：spawn-fcgi这个包原来是属于Lighttpd里的，现在已经可以独立出来使用。

**什么是CGI、FastCGI、PHP-CGI、PHP-FPM、Spawn-FCGI？**

[**http://www.mike.org.cn/articles/what-is-cgi-fastcgi-php-fpm-spawn-fcgi/**](http://www.mike.org.cn/articles/what-is-cgi-fastcgi-php-fpm-spawn-fcgi/)

安装时配置项太多，可以参考帮助./configure --help，具体实例请看下面：

./configure --prefix=/home/apps/php-5.5.10 --with-config-file-path=/home/apps/php-5.5.10/etc \

--with-iconv-dir=/usr/local \

--with-mcrypt=/usr/local \

--with-mhash=/usr/local/ --with-pcre-regex=/usr/local \

--with-zlib=/usr/local/ \

--with-jpeg-dir=/usr/local/ --with-png-dir=/usr/local/ \

--with-freetype-dir=/usr/local/ --with-mysql=/usr/local/mysql \

--with-mysqli=/usr/local/mysql/bin/mysql\_config \

--with-openssl --enable-fpm --enable-zip \

--enable-xml --enable-mbstring --enable-zip --enable-opcache \

--with-curl --enable-sockets --enable-soap --with-gd --enable-gd-native-ttf \

--enable-zip --enable-exif --enable-ftp --with-pdo-mysql

--enable-mbstring PHP多字节字符串处理函数mbstring函数库

## OPCODE缓存

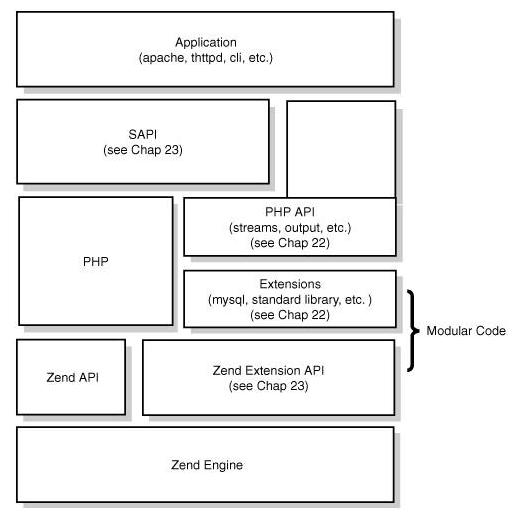
SAPI:Server Application Programming Interface 服务器端应用编程端口，它提供了一个接口，使得PHP可以和其他应用进行交互数据。

SAPI提供了一个和外部通信的接口， 对于PHP5.2，默认提供了很多种SAPI， 常见的给apache的mod\_php5，CGI，给IIS的ISAPI，还有Shell的CLI

PHP内核用来处理请求、文件流、错误处理等相关操作；

Zend引擎（ZE）用以将源文件转换成机器语言，然后在虚拟机上运行它

比如，我们需要mysql扩展来连接MySQL数据库； 当ZE执行程序时可能会需要连接若干扩展，这时ZE将控制权交给扩展，等处理完特定任务后再返还；最后，ZE将程序运行结果返回给PHP内核，它再将结果传送给SAPI层，最终输出到浏览器上。



引擎在解析PHP代码的时候通常是分为两个部分，编译和执行：

编译阶段：引擎把PHP代码转换成op code中间代码

执行阶段：引擎解释并执行编译阶段产生的op code

PHP生命周期

请求--->.php--->词典扫描--->解析--->创建Opcode--->处理Opcode--->响应

即使该PHP脚本的内容没有任何变化，Zend引擎也必须重新创建该文件的Opcode.

Opcode缓存提高PHP性能

请求--->.php  --->有缓存--->读取已缓存的Opcode--->处理Opcode--->响应  
请求--->.php --->无缓存--->词典扫描--->解析--->创建Opcode--->处理Opcode--->响应

Apc，eAccelerator加速可以提神一倍的性能(需要数据测试)。

Zend Opcache 已经集成在了PHP 5.5里面，编译安装PHP5.5的时候加上--enable-opcache就行了。

## PHP扩展

一、安装libevent开发包

yum install libevent-devel -y

二、安装Memcached

我们从http://memcached.org/获取Memcached安装包memcached-1.4.15.tar.gz。

1、配置（通过参数prefix指定安装目录）

./configure --prefix=/home/apps/mc-1.4.17 --enable-sasl

2、编译和安装

make

make install

三、安装libmemcached

从http://libmemcached.org/下载安装包libmemcached-1.0.15.tar.gz。

配置、编译和安装（–with-memcached指定Memcached执行程序）

./configure --with-memcached=/usr/local/memcached/bin/memcached

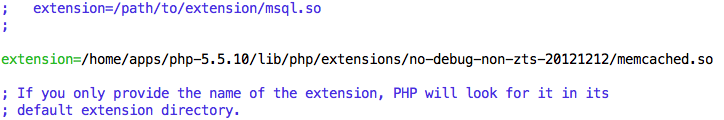
make

make install

四、安装PHP memcached扩展

/home/apps/php-5.5.10/bin/pecl install memcached

修改php.ini文件，添加记录



http://starnat.me/2013/08/08/memcached-installation-and-configuration-with-php-extension-on-centos/

php.ini:

zend\_extension=/home/apps/php-5.5.10/lib/php/extensions/no-debug-non-zts-20121212/opcache.so

[opcache]

; Determines if Zend OPCache is enabled

opcache.enable=1

; Determines if Zend OPCache is enabled for the CLI version of PHP

opcache.enable\_cli=1

; The OPcache shared memory storage size.

opcache.memory\_consumption=512

; The amount of memory for interned strings in Mbytes.

;opcache.interned\_strings\_buffer=4

; The maximum number of keys (scripts) in the OPcache hash table.

; Only numbers between 200 and 100000 are allowed.

opcache.max\_accelerated\_files=80000

设置expose\_php = Off

这样就避免了输出类似  
X-Powered-By: PHP/5.2.4

## 开机启动

/etc/rc.d/rc.local：将开机要启动的可执行命令行添加到文件中即可如：

开机启动mysql，nginx及php：

*/usr/local/mysql/bin/mysqld\_safe &*

*/usr/local/nginx/sbin/nginx*

*/usr/local/php/sbin/php-fpm*

## 动态库引用

/etc/ld.so.conf 将要引入有动态库目录加入到该

ldconfig 将新加入的动态库目录生效

如：

/usr/local/lib

/usr/local/mysql/lib

/usr/local/mhash/lib

## 优化及配置

/Etc/Sysctl.conf

vm.swappiness = 1

net.ipv4.tcp\_max\_syn\_backlog = 65536

net.core.netdev\_max\_backlog = 32768

net.core.somaxconn = 32768

net.ipv4.tcp\_timestamps = 0

net.ipv4.tcp\_synack\_retries = 2

net.ipv4.tcp\_syn\_retries = 2

net.ipv4.tcp\_tw\_recycle = 1

#net.ipv4.tcp\_tw\_len = 1

net.ipv4.tcp\_tw\_reuse = 1

net.ipv4.tcp\_mem = 94500000 915000000 927000000

net.ipv4.tcp\_max\_orphans = 3276800

net.ipv4.ip\_conntrack\_max= 6553500

net.ipv4.tcp\_fin\_timeout = 30

#net.ipv4.tcp\_keepalive\_time = 120

net.ipv4.ip\_local\_port\_range = 1024 65535

fs.file-max = 6553500

## PHP配置

502错误：

request\_terminate\_timeout：PHP脚本的最大执行时间

request\_terminate\_timeout = 60

PHP-CGI 进程 CPU 100% 与 file\_get\_contents 函数的关系

http://blog.s135.com/file\_get\_contents/

<http://grb12508.blog.163.com/blog/static/2737845820119101022211/>

;动态配置  
pm = dynamic  
;每个进程的子线程  
pm.max\_children = 25  
;缺省启动进程  
pm.start\_servers = 12  
;最小启动进程数  
pm.min\_spare\_servers = 5  
;最大启动进程数  
pm.max\_spare\_servers = 20  
;最大请求数  
pm.max\_requests = 500  
;超时关闭请求，  
request\_terminate\_timeout = 10s  
;超时日志，可以找出什么php运行速度慢  
request\_slowlog\_timeout = 2s  
;慢速php的日志  
slowlog = /var/log/fpm.$pool.log.slow

/etc/profile

ulimit -SHn 262144

## 参考：

Unbutu 10.04(X64)下编译安装LNMP平台(Nginx1.0+PHP5.3.6+Mysql5.5.11)

<http://www.mike.org.cn/articles/unbutu-compile-install-lnmp>

Nginx 0.8.x + PHP 5.2.13（FastCGI）搭建胜过Apache十倍的Web服务器（第6版）

<http://blog.s135.com/nginx_php_v7/>

http://www.howtocn.org/nginx:%E9%85%8D%E7%BD%AE%E4%B8%80%E4%B8%AA%E9%AB%98%E6%80%A7%E8%83%BD%E7%9A%84nginx\_fastcgi%E6%9C%8D%E5%8A%A1%E5%99%A8