fpm模式安装php-5.4.40及xcache

环境介绍：

操作系统：CentOS release 6.7 (Final)

所需软件：php-5.4.40.tar.bz2,xcache-3.2.0.tar.bz2

开发环境：Development Tools, Server Platform Development

httpd版本：httpd-2.4.25

数据库版本：mariadb-5.5.43

目标：php以fpm工作为独立的守护进程

0、解决依赖关系

(1）配置好yum源（系统安装源及epel源）

使用阿里云镜像站点作为epel源，安装如下：

# wget <https://mirrors.aliyun.com/epel/epel-release-latest-6.noarch.rpm>

# rpm -ivh epel-release-latest-6.noarch.rpm

（2）安装相关依赖包

# yum -y install bzip2-devel libmcrypt-devel libxml2-devel

1、下载源码安装包

php官网下载：<http://php.net/downloads.php>

xcache官网下载：wget <http://xcache.lighttpd.net/pub/Releases/3.2.0/xcache-3.2.0.tar.bz2>

2、编译安装php-5.4.40

src]# tar -xf php-5.4.40.tar.bz2

src]# cd php-5.4.40

php-5.4.40]# ./configure --prefix=/usr/local/php --with-mysql=/usr/local/mysql/ --with-mysqli=/usr/local/mysql/bin/mysql\_config --enable-mbstring --with-freetype-dir --with-jpeg-dir --with-png-dir --with-zlib --with-libxml-dir --enable-xml --enable-sockets --with-mcrypt --with-bz2 --with-config-file-path=/etc/php/php.ini --with-config-file-scan-dir=/etc/php.d/ --enable-fpm

php-5.4.40]# make

php-5.4.40]# make test

php-5.4.40]# make install

--prefix：默认安装目录

--with-mysql：指明mysql安装路径,标准路径：/usr/local/mysql

--with-openssl：指明使用openssl功能

--with-mysqli：mysql的另外一种访问接口，在某些嵌入式访问中可能用到，不重要，此次可以不使用

--enable-mbstring：指明php支持多字节文本

--with-freetype-dir：指明freetype字库目录，可能需要事先安装freetype-devel包。

--with-jpeg-dir：指明支持jpeg格式图片生成

--with-png-dir：指明支持png格式图片

--with-zlib：指明支持zlib

该模块使您可以通过使用gzip压缩文件（以及未压缩文件，但不包含套接字）的大多数文件系统功能的版本，透明地读取和写入gzip（.gz）压缩文件。

--with-libxml：指明xml lib库，依赖libxml2-devel包，需事先安装

--enable-xml：启用支持xml文档的处理

--enable-sockets：启用sockets功能

--enable-fpm：启用fpm

--with-mcrypt：依赖libmcrypt-devel包，需事先安装，在epel源中才有。

mcrypt 是php里面重要的加密支持扩展库，linux环境下该库在默认情况下不开启

--with-config-file-path：指明主配置文件路径

--with-config-file-sca：指明扫描置顶路径下所有以.ini结尾配置文件

--with-bz2：指明支持bz2，依赖bzip2-devel包，需事先安装

bzip2 函数用于透明地读写 bzip2（.bz2）压缩文件。

--enable-maintainer-zts：启用php支持多线程模式，此参数表明开启线程安全。

说明：

（1）、这里为了支持apache的worker或event这两个MPM，编译时使用了—enable-maintainer-zts选项。

（2）、如果使用PHP5.3以上版本，为了链接MySQL数据库，可以指定mysqlnd，这样在本机就不需要先安装MySQL或MySQL开发包了。mysqlnd从php 5.3开始可用，可以编译时绑定到它（而不用和具体的MySQL客户端库绑定形成依赖），但从PHP 5.4开始它就是默认设置了。

# ./configure --with-mysql=mysqlnd --with-pdo-mysql=mysqlnd --with-mysqli=mysqlnd

3、为php提供配置文件

php-5.4.40]# mkdir /etc/php

php-5.4.40]# cp php.ini-production /etc/php/php.ini

4、配置php-fpm

（1）为php-fpm提供SysV init脚本，并将其添加至服务列表

php-5.4.40]# cp sapi/fpm/init.d.php-fpm /etc/rc.d/init.d/php-fpm

php-5.4.40]# chmod +x /etc/rc.d/init.d/php-fpm

php-5.4.40]# chkconfig --add php-fpm

php-5.4.40]# chkconfig php-fpm on

（2）为php-fpm提供配置文件

~]# cp /usr/local/php/etc/php-fpm.conf.default /usr/local/php/etc/php-fpm.conf

（3）编辑php-fpm的配置文件

~]# vim /usr/local/php/etc/php-fpm.conf

配置fpm的相关选项为你所需要的值，并启用pid文件

pid = /usr/local/php/var/run/php-fpm.pid

pm.max\_children = 50

pm.start\_servers = 5

pm.min\_spare\_servers = 2

pm.max\_spare\_servers = 5

5、启动php-fpm

~]# service php-fpm start

Starting php-fpm done

使用如下命令来验正（如果此命令输出有中几个php-fpm进程就说明启动成功了）：

# ps aux | grep php-fpm

默认情况下，fpm监听在127.0.0.1的9000端口，也可以使用如下命令验正其是否已经监听在相应的套接字。

~]# netstat -tnlp | grep php-fpm

tcp 0 0 127.0.0.1:9000 0.0.0.0:\* LISTEN 52888/php-fpm

6、配置httpd-2.4.9

（1）启用httpd的相关模块

在Apache httpd 2.4以后已经专门有一个模块针对FastCGI的实现，此模块为mod\_proxy\_fcgi.so，它其实是作为mod\_proxy.so模块的扩充，因此，这两个模块都要加载

LoadModule proxy\_module modules/mod\_proxy.so

LoadModule proxy\_fcgi\_module modules/mod\_proxy\_fcgi.so

（2）配置虚拟主机支持使用fcgi

在相应的虚拟主机中添加类似如下两行。

ProxyRequests Off

ProxyPassMatch ^/(.\*\.php)$ fcgi://127.0.0.1:9000/PATH/TO/DOCUMENT\_ROOT/$1

http://www.magedu.com/admin/index.php

/web/host1/admin/index.php

fcgi://127.0.0.1:9000/web/hosts/admin/index.php

例如：

<VirtualHost \*:80>

DocumentRoot "/www/magedu.com"

ServerName magedu.com

ServerAlias www.magedu.com

ProxyRequests Off

ProxyPassMatch ^/(.\*\.php)$ fcgi://127.0.0.1:9000/www/magedu.com/$1

<Directory "/www/magedu.com">

Options none

AllowOverride none

Require all granted

</Directory>

</VirtualHost>

ProxyRequests Off：关闭正向代理

ProxyPassMatch：把以.php结尾的文件请求发送到php-fpm进程，php-fpm至少需要知道运行的目录和URI，所以这里直接在fcgi://127.0.0.1:9000后指明了这两个参数，其它的参数的传递已经被mod\_proxy\_fcgi.so进行了封装，不需要手动指定。

~]# vim /etc/httpd/httpd.conf

ProxyRequests Off

ProxyPassMatch ^/(.\*\.php)$ fcgi://127.0.0.1:9000/usr/local/apache/htdocs/$1

（3）编辑apache配置文件httpd.conf，让apache能识别php格式的页面，并支持php格式的主页

# vim /etc/httpd/httpd.conf

1、添加如下二行

AddType application/x-httpd-php .php

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

2、定位至DirectoryIndex index.html

修改为：

DirectoryIndex index.php index.html

补充：Apache httpd 2.4以前的版本中，要么把PHP作为Apache的模块运行，要么添加一个第三方模块支持PHP-FPM实现。

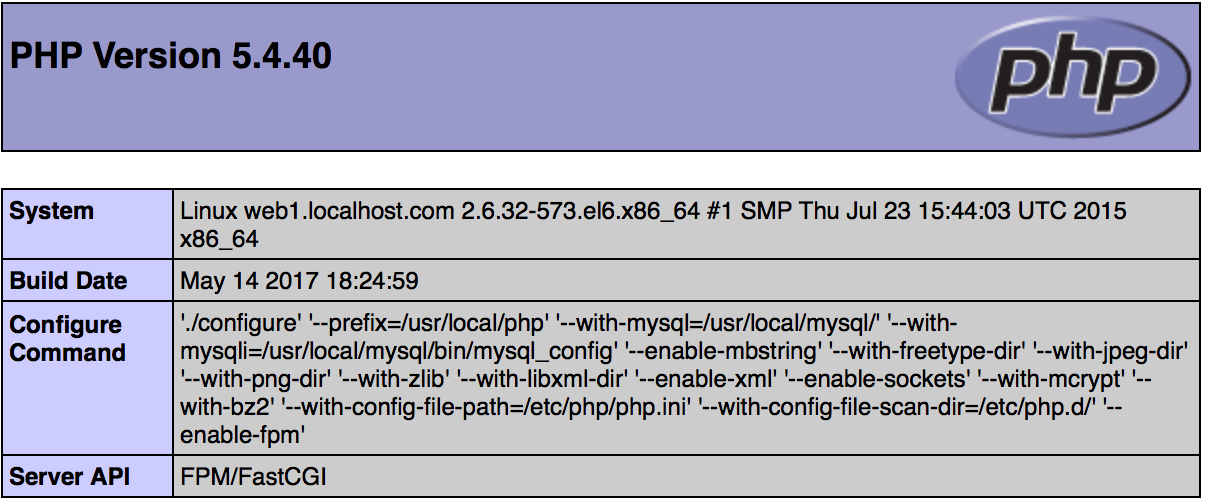
7、创建测试文件并测试

~]# vim /usr/local/apache/htdocs/index.php

<?php

phpinfo();

?>

~]# service httpd reload

8、部署phpMyAdmin进行测试

（1）下载phpMyAdmin

src]# wget https://files.phpmyadmin.net/phpMyAdmin/4.0.10.20/phpMyAdmin-4.0.10.20-all-languages.zip

（2）解压

src]# unzip phpMyAdmin-4.0.10.20-all-languages.zip

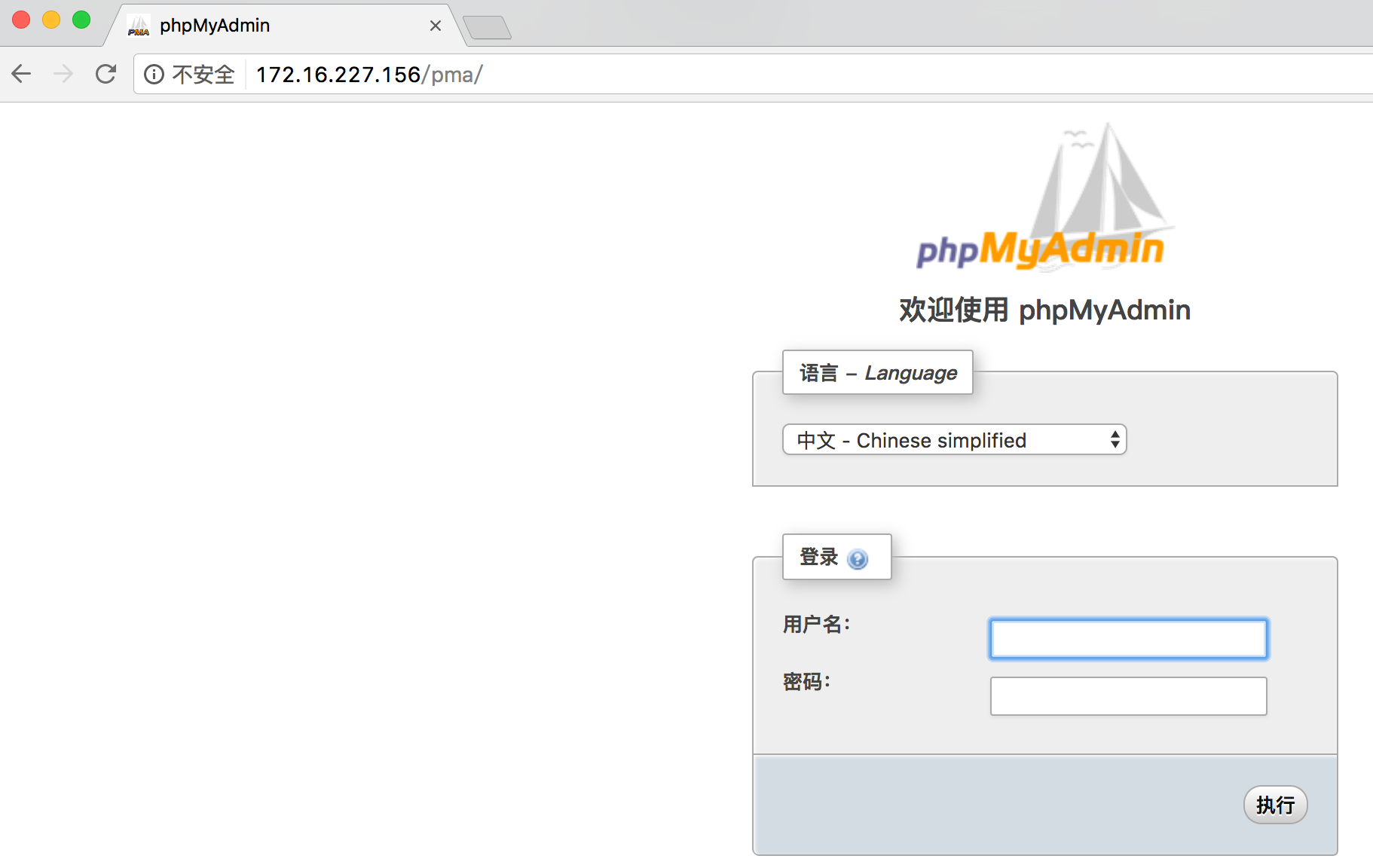
（3）将解压后的phpMyAdmin移动到/usr/local/apache/htdocs/目录下，并重命名为pma

src]# mv phpMyAdmin-4.0.10.20-all-languages /usr/local/apache/htdocs/pma

（4）准备配置文件

src]# cd /usr/local/apache/htdocs/pma

pma]# cp config.sample.inc.php config.inc.php

（5）访问测试

9、使用ab工具测网页性能

~]# ab -n 1000 -c 50 http://172.16.227.156/pma/index.php

This is ApacheBench, Version 2.3 <$Revision: 1757674 $>

Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/

Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 172.16.227.156 (be patient)

Completed 100 requests

Completed 200 requests

Completed 300 requests

Completed 400 requests

Completed 500 requests

Completed 600 requests

Completed 700 requests

Completed 800 requests

Completed 900 requests

Completed 1000 requests

Finished 1000 requests

Server Software: Apache/2.4.25

Server Hostname: 172.16.227.156

Server Port: 80

Document Path: /pma/index.php

Document Length: 8714 bytes

Concurrency Level: 50

Time taken for tests: 29.161 seconds

Complete requests: 1000

Failed requests: 0

Total transferred: 10153596 bytes

HTML transferred: 8714000 bytes

Requests per second: 34.29 [#/sec] (mean) <—与使用模块时差别不大，因为在同一台主机的缘故？

Time per request: 1458.071 [ms] (mean)

Time per request: 29.161 [ms] (mean, across all concurrent requests)

Transfer rate: 340.03 [Kbytes/sec] received

Connection Times (ms)

min mean[+/-sd] median max

Connect: 0 5 24.6 0 148

Processing: 647 1437 221.9 1471 2045

Waiting: 628 1395 220.3 1428 2029

Total: 749 1442 216.9 1475 2045

Percentage of the requests served within a certain time (ms)

50% 1475

66% 1522

75% 1562

80% 1588

90% 1647

95% 1731

98% 1875

99% 1925

100% 2045 (longest request)

10、安装xcache-3.2.0进行php加速

src]# tar xf xcache-3.2.0.tar.bz2

src]# cd xcache-3.2.0

xcache是PHP的模块所以它没有configure命令，需使用phpize命令来生成configure命令。

xcache-3.2.0]# /usr/local/php/bin/phpize

Configuring for:

PHP Api Version: 20100412

Zend Module Api No: 20100525

Zend Extension Api No: 220100525

xcache-3.2.0]# ./configure --enable-xcache --with-php-config=/usr/local/php/bin/php-config

--enable-xcache：启用xcache

--with-php-config：指明php配置接口

xcache-3.2.0]# make

xcache-3.2.0]# make install

Installing shared extensions: /usr/local/php/lib/php/extensions/no-debug-non-zts-20100525/

11、配置php支持xcache

xcache-3.2.0]# mkdir /etc/php.d

xcache-3.2.0]# cp xcache.ini /etc/php.d/

xcache-3.2.0]# cd /etc/php.d/

php.d]# vim xcache.ini

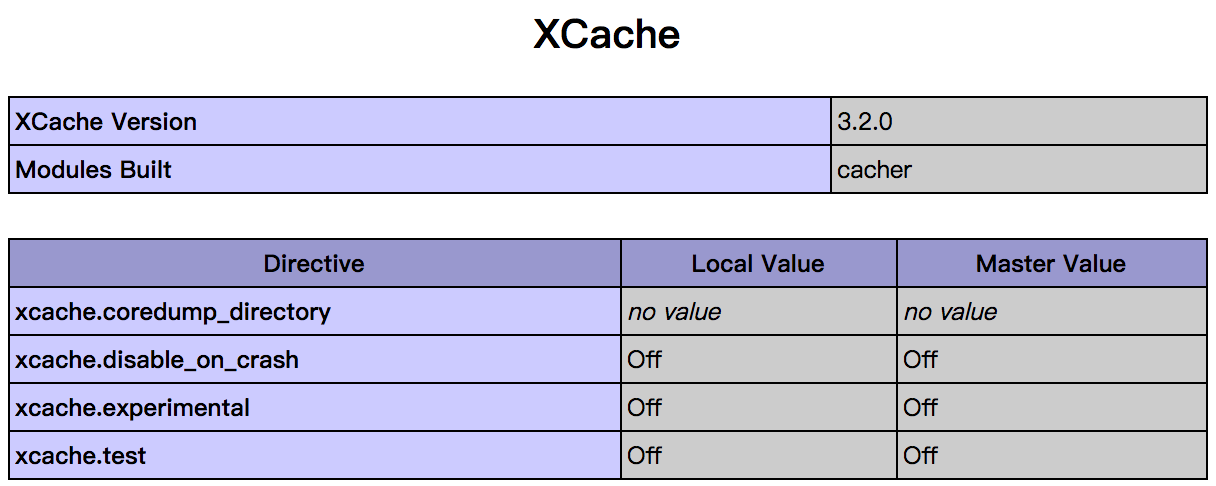
找到extension开头的行，修改为如下行：

extension = /usr/local/php/lib/php/extensions/no-debug-non-zts-20100525/xcache.so

12、重启httpd服务进行测试

]# service php-fpm restart

Gracefully shutting down php-fpm . done

Starting php-fpm done

13、使用ab再次测试

~]# ab -n 1000 -c 50 http://172.16.227.156/pma/index.php

This is ApacheBench, Version 2.3 <$Revision: 1757674 $>

Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/

Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 172.16.227.156 (be patient)

Completed 100 requests

Completed 200 requests

Completed 300 requests

Completed 400 requests

Completed 500 requests

Completed 600 requests

Completed 700 requests

Completed 800 requests

Completed 900 requests

Completed 1000 requests

Finished 1000 requests

Server Software: Apache/2.4.25

Server Hostname: 172.16.227.156

Server Port: 80

Document Path: /pma/index.php

Document Length: 8714 bytes

Concurrency Level: 50

Time taken for tests: 5.607 seconds

Complete requests: 1000

Failed requests: 0

Total transferred: 10153610 bytes

HTML transferred: 8714000 bytes

Requests per second: 178.36 [#/sec] (mean) <—提升不止一点点

Time per request: 280.336 [ms] (mean)

Time per request: 5.607 [ms] (mean, across all concurrent requests)

Transfer rate: 1768.53 [Kbytes/sec] received

Connection Times (ms)

min mean[+/-sd] median max

Connect: 0 1 6.4 0 36

Processing: 32 273 40.2 277 440

Waiting: 25 265 39.8 270 439

Total: 45 275 36.4 278 440

Percentage of the requests served within a certain time (ms)

50% 278

66% 284

75% 289

80% 293

90% 301

95% 308

98% 325

99% 351

100% 440 (longest request)