2.6引入epoll模型

apache采用select模型 稳定性较高 内核逐个扫描 C10K问题

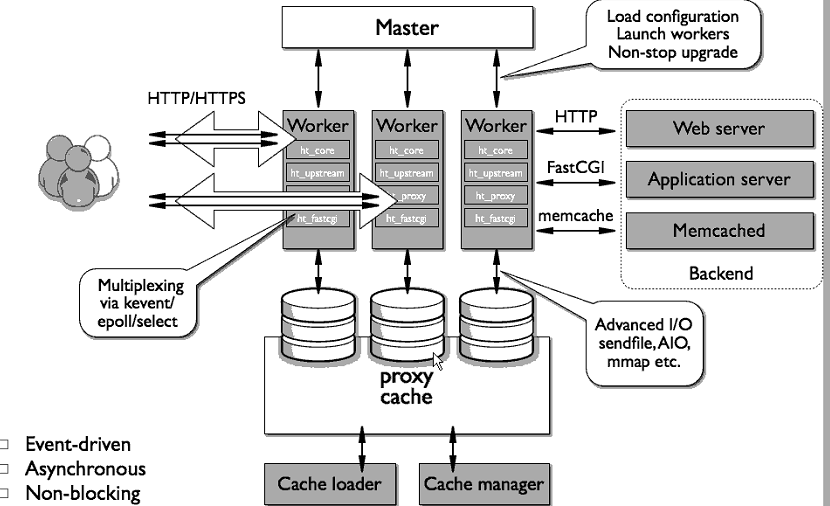
nginx：基于事件通知 通知内核 epoll模型 5-6W

水平触发 通知一次

边缘触发 通知直到反应

原生不支持CGI

需额外编译安装，对静态反应快，对动态反应一般，邮件web反向代理



事件模型

异步（网络IO一般是同步的）

非阻塞

localtion的位置符号作用 匹配符

nginx

Nginx is a free, open-source, high-performance HTTP server and reverse proxy, as well as an IMAP/POP3 proxy server. Igor Sysoev started development of Nginx in 2002, with the first public release in 2004. Nginx now hosts nearly 7.67% (35.5M) of all domains worldwide.

Nginx is known for its high performance, stability, rich feature set, simple configuration, and low resource consumption.

Nginx is one of a handful of servers written to address the C10K problem. Unlike traditional servers, Nginx doesn't rely on threads to handle requests. Instead it uses a much more scalable event-driven (asynchronous) architecture. This architecture uses small, but more importantly, predictable amounts of memory under load.

Even if you don't expect to handle thousands of simultaneous requests, you can still benefit from Nginx's high-performance and small memory footprint. Nginx scales in all directions: from the smallest VPS all the way up to clusters of servers.

Nginx powers several high-visibility sites, such as WordPress, Hulu, Github, Ohloh, SourceForge, WhitePages and TorrentReactor.

功能：

web服务器

web reverse proxy

smtp reverse proxy

LNMP fastcgi, lighttpd (GNU, GUI, Gnome)

LNMP (cache, apc)

Corosync + ningx

Installing the nginx

yum install gcc openssl-devel pcre-devel zlib-devel

# groupadd -r nginx

# useradd -r -g nginx -s /bin/false -M nginx

./configure \

--prefix=/usr \

--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/nginx.pid \

--lock-path=/var/lock/nginx.lock \

--user=nginx \

--group=nginx \

--with-http\_ssl\_module \

--with-http\_flv\_module \

--with-http\_stub\_status\_module \

--with-http\_gzip\_static\_module \

--http-client-body-temp-path=/var/tmp/nginx/client/ \

--http-proxy-temp-path=/var/tmp/nginx/proxy/ \

--http-fastcgi-temp-path=/var/tmp/nginx/fcgi/ \

--http-uwsgi-temp-path=/var/tmp/nginx/uwsgi \

--http-scgi-temp-path=/var/tmp/nginx/scgi \

--with-pcre

/usr/html

make && make install

关于配置选项的简单说明：

--prefix=<path> - The path relative to which all other Nginx paths will resolve. If not specified, defaults to /usr/local/nginx.

--sbin-path=<path> - The path to the nginx executable. Only used for installation. If not specified defaults to <prefix>/sbin/nginx.

--conf-path=<path> - The default location of nginx.conf if no -c parameter is provided. If not provided, defaults to <prefix>/conf/nginx.conf.

--pid-path=<path> - The path to nginx.pid, if not set via the "pid" directive in nginx.conf. If not provided, defaults to <prefix>/logs/nginx.pid.

--error-log-path=<path> - The location of the error log if not set via the "error\_log" in nginx.conf. If not set, defaults to <prefix>/logs/error.log.

--http-log-path=<path> - The location of the access log if not set via the "access\_log" directive in nginx.conf. If not set, defaults to <prefix>/logs/access.log.

--user=<user> - The default user that nginx will run as if not set in nginx.conf via the "user" directive. If not set, defaults to "nobody".

--group=<group> - The default group that nginx will run under if not set via the "user" directive in nginx.conf. If not set defaults to "nobody".

--with-http\_ssl\_module - Enable ngx\_http\_ssl\_module. Enables SSL support and the ability to handle HTTPS requests. Requires OpenSSL. On Debian, this is libssl-dev.

--with-http\_flv\_module - Enable ngx\_http\_flv\_module

--http-client-body-temp-path=PATH - Set path to the http client request body temporary files. If not set, defaults to <prefix>/client\_body\_temp

--http-proxy-temp-path=PATH - Set path to the http proxy temporary files. If not set, defaults to <prefix>/proxy\_temp

--http-fastcgi-temp-path=PATH - Set path to the http fastcgi temporary files. If not set, defaults to <prefix>/fastcgi\_temp

--lock-path=<path> - The path to the nginx.lock file. If not provided, defaults to <prefix>/logs/nginx.lock.

编译安装的 **启动脚本**：

LNMP.txt有

把内容贴进去 增加权限即可

#vim /etc/rc.d/init.d/nginx

#chmod +x /etc/rc.d/init.d/nginx

#chconfig --add nginx

#service nginx restart

What is FastCGI

FastCGI is a high-speed and scalable interface for communicating with the web server scripting language. FastCGI is supported by many scripting languages, including php, if it is compiled with the option - --enable-fastcgi.

It is supported by most popular web servers, including Apache (mod\_fastcgi and mod\_fcgid), Zeus, nginx and lighttpd. The main advantage of FastCGI is isolating the dynamic language from the web server. The technology, among other things, allows you to run a web server and dynamic language for the different hosts, which improves scalability and also aids security without a significant loss of productivity.

PHP-FPM works on with any web server that supports FastCGI.

Howto

Make sure libxml2 (and libxml2-devel) is installed, and libevent version is 1.4.12 or later.

for PHP 5.2.x:

$ bzip2 -cd php-5.2.13.tar.bz2 | tar xf -

$ patch -d php-5.2.13 -p1 <php-fpm-0.6~5.2.patch

$ cd php-5.2.13

$ ./buildconf --force

$ ./configure --enable-fastcgi --with-fpm --with-libevent[=path] ...

autorun php-fpm:

$ ln -s /usr/local/sbin/php-fpm /etc/init.d/php-fpm

$ /usr/sbin/update-rc.d -f php-fpm defaults

for PHP 5.3.x:

$ cd php-5.3.x

$ svn co http://svn.php.net/repository/php/php-src/trunk/sapi/fpm sapi/fpm

$ ./buildconf --force

$ ./configure --enable-fpm ...your other configure options, etc...

$ make && make install

Edit /etc/php-fpm.conf

Run php-fpm start (probably in your $PATH). Check logfile /var/log/php-fpm.log for details if needed.

PHP-FPM (FastCGI Process Manager) is an alternative PHP FastCGI implementation with some additional features useful for sites of any size, especially busier sites.

for PHP 5.3.3

PHP-FPM is now included in PHP core as of PHP 5.3.3. Make sure libxml2 (and libxml2-devel) is installed, and libevent version is 1.4.12 or later, and libiconv.

libevent

The libevent API provides a mechanism to execute a callback function when a specific event occurs on a file descriptor or after a timeout has been reached. Furthermore, libevent also support callbacks due to signals or regular timeouts.

libevent is meant to replace the event loop found in event driven network servers. An application just needs to call event\_dispatch() and then add or remove events dynamically without having to change the event loop.

Currently, libevent supports /dev/poll, kqueue(2), event ports, select(2), poll(2) and epoll(4). The internal event mechanism is completely independent of the exposed event API, and a simple update of libevent can provide new functionality without having to redesign the applications. As a result, Libevent allows for portable application development and provides the most scalable event notification mechanism available on an operating system. Libevent can also be used for multi-threaded applications.

# tar zxvf libevent-1.4.14b-stable.tar.gz

# cd libevent-1.4.14b-stable

# ./configure

# make && make install

# make verify

libiconv

For historical reasons, international text is often encoded using a language or country dependent character encoding. With the advent of the internet and the frequent exchange of text across countries - even the viewing of a web page from a foreign country is a "text exchange" in this context -, conversions between these encodings have become important. They have also become a problem, because many characters which are present in one encoding are absent in many other encodings. To solve this mess, the Unicode encoding has been created. It is a super-encoding of all others and is therefore the default encoding for new text formats like XML.

Still, many computers still operate in locale with a traditional (limited) character encoding. Some programs, like mailers and web browsers, must be able to convert between a given text encoding and the user's encoding. Other programs internally store strings in Unicode, to facilitate internal processing, and need to convert between internal string representation (Unicode) and external string representation (a traditional encoding) when they are doing I/O. GNU libiconv is a conversion library for both kinds of applications.

# tar zxvf libiconv-1.13.1.tar.gz

# cd libiconv-1.13.1

# ./configure

# make && make install

libmcrypt

MCrypt is a replacement for the old crypt() package and crypt(1) command, with extensions. It allows developers to use a wide range of encryption functions, without making drastic changes to their code. It allows users to encrypt files or data streams without having to be cryptographers. Above all, it allows you to have some really neat code on your machine. :)

The companion to MCrypt is Libmcrypt, which contains the actual encryption functions themselves, and provides a standardized mechanism for accessing them.

# tar zxvf libmcrypt-2.5.8.tar.gz

# cd libmcrypt-2.5.8

# ./configure

# make && make install

# ldconfig -v

# cd libltdl

# ./configure --with-gmetad --enable-gexec --enable-ltdl-install

# make && make install

mhash

Mhash is a free (under GNU Lesser GPL) library which provides a uniform interface to a large number of hash algorithms. These algorithms can be used to compute checksums, message digests, and other signatures.

# tar jxvf mhash-0.9.9.9.tar.bz2

# cd mhash-0.9.9.9

# ./configure

# make && make install

# ln -sv /usr/local/lib/libmcrypt\* /usr/lib/

# ln -sv /usr/local/lib/libmhash.\* /usr/lib/

php-5.3.6

# tar jxvf php-5.3.6.tar.bz2

# cd jxvf php-5.3.6

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

--with-iconv=/usr/local

# make ZEND\_EXTRA\_LIBS='-liconv'

# make install

# cp php.ini-production /usr/local/php/etc/php.ini

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

#

启动fastcgi：

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

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

启用如下选项：

pm.max\_children = 50

pm.start\_servers = 5

pm.min\_spare\_servers = 2

pm.max\_spare\_servers = 8

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

#

编译php跟apache起工作：

./configure --prefix=/usr/local/php4httpd --with-mysql=/usr/local/mysql --with-openssl --with-apxs2=/usr/local/apache/bin/apxs --with-mysqli=/usr/local/mysql/bin/mysql\_config --enable-mbstring --with-freetype-dir --with-jpeg-dir --with-png-dir --with-zlib-dir --with-libxml-dir=/usr --enable-xml --with-mhash --with-mcrypt --with-bz2 --with-curl

接下来整合nginx和php5

编辑/etc/nginx/nginx.conf，启用如下选项：

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;

}

FastCGI Example

First thing, I recommend keeping all your typical FCGI settings in a single file and importing them.

For example you might have an /etc/nginx/fastcgi.conf (or /etc/nginx/fastcgi\_params: installed by default on debian) file that looks like this:

#vim fastcgi\_params

fastcgi\_param GATEWAY\_INTERFACE CGI/1.1;

fastcgi\_param SERVER\_SOFTWARE nginx;

fastcgi\_param QUERY\_STRING $query\_string;

fastcgi\_param REQUEST\_METHOD $request\_method;

fastcgi\_param CONTENT\_TYPE $content\_type;

fastcgi\_param CONTENT\_LENGTH $content\_length;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

fastcgi\_param SCRIPT\_NAME $fastcgi\_script\_name;

fastcgi\_param REQUEST\_URI $request\_uri;

fastcgi\_param DOCUMENT\_URI $document\_uri;

fastcgi\_param DOCUMENT\_ROOT $document\_root;

fastcgi\_param SERVER\_PROTOCOL $server\_protocol;

fastcgi\_param REMOTE\_ADDR $remote\_addr;

fastcgi\_param REMOTE\_PORT $remote\_port;

fastcgi\_param SERVER\_ADDR $server\_addr;

fastcgi\_param SERVER\_PORT $server\_port;

fastcgi\_param SERVER\_NAME $server\_name;

并在所支持的主页面格式中添加php格式的主页，类似如下：

location / {

root html;

index index.php index.html index.htm;

}

而后重启nginx。

# service nginx restart

为FCGI设定缓存：

http {

fastcgi\_cache\_path /www/cache levels=1:2

keys\_zone=fcgicache:10m

inactive=5m;

server {

server\_name wwww.magedu.com;

...

location / {

...

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_cache fcgicache;

fastcgi\_cache\_valid 200 302 1h;

fastcgi\_cache\_valid 301 1d;

fastcgi\_cache\_valid any 1m;

fastcgi\_cache\_min\_uses 1;

fastcgi\_cache\_use\_stale error timeout invalid\_header http\_500;

}

}

}

xcache安装配置：

# tar xf xcache-1.3.2.tar.gz

# cd xcache-1.3.2

# /usr/local/php/bin/phpize

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

# make && make install

安装结束时，会出现类似如下行：

Installing shared extensions: /usr/local/php/lib/php/extensions/no-debug-non-zts-20090626/

# cat xcache.ini >> /usr/local/php/lib/php.ini

接下来编辑/usr/local/php/lib/php.ini，找到zend\_extension开头的行，修改为如下行：

zend\_extension = /usr/local/php/lib/php/extensions/no-debug-non-zts-20090626/xcache.so

注意：如果php.ini文件中有多条zend\_extension指令行，要确保此新增的行排在第一位。

各配置选项说明：

xcache.admin.user string

auth name.

xcache.admin.pass string

Should be md5($your\_password), or empty to disable administration.

xcache.test string

Turn on to enable testing functionals. It will be explained where the option is needed.

xcache.coredump\_directory string

Directory to save core dump on crash (SIGSEGV SIGABRT). Leave it empty to disable or something like "/tmp/phpcore/" to enable. Make sure it's writable by php (without checking open\_basedir).

xcache.admin.enable\_auth string

Disable XCache builtin http authentication if you plan on handling authentication yourself. Be aware that any vhost users can set up admin page, if builtin http auth is disabled, they can access the page with out any authentication. So it is suggested that you disable mod\_auth for XCache admin pages instead of disabling XCache builtin auth. This option is 1.2.x only since 1.2.1

xcache.cacher boolean

Enable or disable opcode cacher. Not available if xcache.size is 0.

xcache.size int

0 to disable, non 0 to enable. Check if your system mmap allows.

xcache.count int

Specify how many chunks to split the cache. see SplittedCache

xcache.slots size

Just a "slots" hint for hash, you can always store count(items) > slots. It can be (count(items) \* n) where n is 0.2 to 1, or leave it as is. More slots means faster searching the cache but take more memory.

xcache.ttl seconds

Ttl (Time To Live) value for the php entry (cached opcodes of a file), 0=forever.

xcache.gc\_interval seconds

Garbage collection interval.

xcache.var\_size int

xcache.var\_count int

xcache.var\_slots size

Same as above, but for variable data.

xcache.var\_ttl seconds

Default ttl for variables api, xcache\_(get|set|inc|dec) etc.

xcache.var\_maxttl seconds

A longer ttl when using variables api is limited to below max ttl.

xcache.var\_gc\_interval seconds

Garbage collection interval for variables api.

xcache.readonly\_protection boolean

If ReadonlyProtection is turned on, it will be a bit slower, but much safer. This option isn't available for /dev/zero.

xcache.mmap\_path string

for \*nix, xcache.mmap\_path is a file path, not directory. for win32, xcache.mmap\_path is anonymous map name, not a file path. Use something like "/tmp/xcache" if you want to turn on ReadonlyProtection, 2 group of php won't share the same /tmp/xcache.

安装Memcache的PHP扩展

①安装PHP的memcache扩展

# tar xf memcache-2.2.5.tgz

# cd memcache-2.2.5

/usr/local/php/bin/phpize

# ./configure --with-php-config=/usr/local/php/bin/php-config --enable-memcache

# make && make install

上述安装完后会有类似这样的提示：

Installing shared extensions: /usr/local/php/lib/php/extensions/no-debug-non-zts-20090626/

②编辑/usr/local/php/lib/php.ini，在“动态模块”相关的位置添加如下一行来载入memcache扩展：

extension=/usr/local/php/lib/php/extensions/no-debug-non-zts-20090626/memcache.so

而后对memcached功能进行测试，在网站目录中建立测试页面test.php，添加如下内容：

<?php

$mem = new Memcache;

$mem->connect("127.0.0.1", 11211);

$mem->set('mykey', 'Hello World', 0, 600);

$val = $mem->get('mykey');

echo "$val is from memcached.";

?>

如果有输出“Hello World is from memcached.”，则表明memcache已经能够正常工作。

使用libmemcached的客户端工具:

访问memcached的传统方法是使用基于perl语言开发的Cache::memcached模块，这个模块在大多数perl代码中都能良好的工作，但也有着众所周知的性能方面的问题。libMemcached则是基于C语言开发的开源的C/C++代码访问memcached的库文件，同是，它还提供了数个可以远程使用的memcached管理工具，如memcat, memping，memstat，memslap等。

编译安装libmemcached

# tar xf libmemcached-1.0.2.tar.gz

# cd libmemcached-1.0.2

# ./configure

# make && make install

# ldconfig

客户端工具

# memcat --servers=127.0.0.1:11211 mykey

# memping

# memslap

# memstat

Secure your upload directory!!

Too many example configs fail to secure the "uploads" directory of the application. Remember that if someone can upload a file named xyz.php and the uploads dir is publically accessible then you have given the attacker an easy way to insert PHP onto your site...

So if your app has an upload dir "/images/" then insert if ($uri !~ "^/images/") before fastcgi\_pass, as so:

location ~ \.php$ {

...

...

if ($uri !~ "^/images/") {

fastcgi\_pass 127.0.0.1:9000;

}

}

www.magedu.com/images/logo.jpg --> http://img.magedu.com/images/logo.jpg

rewirte "/images/\(.\*\.jpg\)" http://img.magedu.com/images/$1

PDO\_MYSQL

PDO\_MYSQL is a driver that implements the PHP Data Objects (PDO) interface to enable access from PHP to MySQL 3.x, 4.x and 5.x databases.

PDO\_MYSQL will take advantage of native prepared statement support present in MySQL 4.1 and higher. If you're using an older version of the mysql client libraries, PDO will emulate them for you.

eAccelerator

eAccelerator is a free open-source PHP accelerator & optimizer. It increases the performance of PHP scripts by caching them in their compiled state, so that the overhead of compiling is almost completely eliminated. It also optimizes scripts to speed up their execution. eAccelerator typically reduces server load and increases the speed of your PHP code by 1-10 times.

# tar jxvf eaccelerator-0.9.6.1.tar.bz2

# cd eaccelerator-0.9.6.1

# /usr/local/php/bin/phpize

# ./configure \

--enable-eaccelerator=shared \

--with-php-config=/usr/local/php/bin/php-config

# make

# make install

# vim /usr/local/php/etc/php.ini

添加如下内容：

extension="eaccelerator.so"

eaccelerator.shm\_size="16"

eaccelerator.cache\_dir="/tmp/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"

# mkdir /tmp/eaccelerator

# chmod 0777 /tmp/eaccelerator

Configuration Options

---------------------

eaccelerator.shm\_size

The amount of shared memory (in megabytes) that eAccelerator will use.

"0" means OS default. Default value is "0".

eaccelerator.cache\_dir

The directory that is used for disk cache. eAccelerator stores precompiled

code, session data, content and user entries here. The same data can be

stored in shared memory also (for more quick access). Default value is

"/tmp/eaccelerator".

eaccelerator.enable

Enables or disables eAccelerator. Should be "1" for enabling or "0" for

disabling. Default value is "1".

eaccelerator.optimizer

Enables or disables internal peephole optimizer which may speed up code

execution. Should be "1" for enabling or "0" for disabling. Default value

is "1".

eaccelerator.debug

Enables or disables debug logging. Should be "1" for enabling or "0" for

disabling. Default value is "0".

eaccelerator.check\_mtime

Enables or disables PHP file modification checking . Should be "1" for

enabling or "0" for disabling. You should set it to "1" if you want to

recompile PHP files after modification. Default value is "1".

eaccelerator.filter

Determine which PHP files must be cached. You may specify the number of

patterns (for example "\*.php \*.phtml") which specifies to cache or not to

cache. If pattern starts with the character "!", it means to ignore files

which are matched by the following pattern. Default value is "" that means

all PHP scripts will be cached.

eaccelerator.shm\_max

Disables putting large values into shared memory by " eaccelerator\_put() "

function. It indicates the largest allowed size in bytes (10240, 10K, 1M).

The "0" disables the limit. Default value is "0".

eaccelerator.shm\_ttl

When eaccelerator fails to get shared memory for new script it removes all

scripts which were not accessed at last "shm\_ttl" seconds from shared

memory. Default value is "0" that means - don't remove any files from

shared memory.

eaccelerator.shm\_prune\_period

When eaccelerator fails to get shared memory for new script it tryes to

remove old script if the previous try was made more then

"shm\_prune\_period" seconds ago. Default value is "0" that means - don't

try to remove any files from shared memory.

eaccelerator.shm\_only

Enables or disables caching of compiled scripts on disk. It has no effect

on session data and content caching. Default value is "0" that means - use

disk and shared memory for caching.

eaccelerator.allowed\_admin\_path

The script paths that are allowed to get admin information and do admin

controls

The Rewrite Module

www.magedu.com/bbs 论坛

www.magedu.com/forum 论坛

page:

图片地址引用：http://172.16.100.4/images/a.jpg

172.16.100.5

http://www.magedu.com/attatch.php?value=111223

http://www.magedu.com/111222/attatch

last

rewrite ^/(attatch)\.php\?value=(.\*)$ /$2/$1 break;

rewrite

rewrite

rewrite

SEO

URL

rewrite

A lot of sites undergo changes, and in some cases complete rewriting. In most cases the earlier contents URLs would have changed, leading to loss of SEO and, of course, inconvenience for older clients. This recipe will help you write simple rewrites so that you can ensure that your new site has all the redirect working.

rewrite指令的语法：

s@\(patt\)ern@\1@

rewrite regex replacement flag

rewrite ^/images/(.\*\.jpg)$ /images2/$1 break;

rewrite ^/abc/.\*$ /$1/abc/ last

http://172.16.100.1/images2/logo.jpg

location / {

rewrite ^/images/.\*\.jpg$ /images/b.jpg break;

}

http://172.16.100.1/images/b.jpg

Flags can be any of the following:

last - completes processing of current rewrite directives and restarts the process (including rewriting) with a search for a match on the URI from all available locations.

break - completes processing of current rewrite directives and non-rewrite processing continues within the current location block only.

redirect - returns temporary redirect with code 302; it is used if the substituting line begins with http://

permanent - returns permanent redirect with code 301

set指令

语法：set variable value

应用环境: server, location, if

为变量设定值；可以是自定义的变量；

一、设置一个简单的URL重写：

比如，某网站原有的论坛访问路径为/forum/，但后来根据要求需要更改为/bbs，于是，就可以通过下面的方法实现：

rewrite ^/forum/?$ /bbs/ permanent;

http://172.16.100.1/forum/

1、if指令：

语法: if (condition) { ... }

应用环境: server, location

条件:

1、变量名; false values are: empty string ("", or any string starting with "0";)

2、对于变量进行的比较表达式，可使用=或!=进行测试;

3、正则表达式的模式匹配:

~ 区分大小的模式匹配

~\* 不区分字母大小写的模式匹配

!~ 和 !~\* 分别对上面的两种测试取反

4、测试文件是否存在-f或!-f

5、测试目录是否存在-d或!-d

6、测试目录、文件或链接文件的存在性-e或!-e

7、检查一个文件的执行权限-x或!-x

在正则表达式中，可以使用圆括号标记匹配到的字符串，并可以分别使用$1,$2,...,$9进行引用；

例如：

判断用户的浏览器类型：

if ($http\_user\_agent ~\* MSIE) {

rewrite ^(.\*)$ /msie/$1 break;

}

wap.magedu.com

if ($http\_user\_agent ~\* opera) {

rewrite ^(.\*)$ /opera/$1 break;

}

如果用户请求的页面不存在，实现自定义跳转：

if (!-f $request\_filename) {

rewrite ^(/.\*)$ /rewrite.html permanent;

}

实现域名跳转

server

{

listen 80;

server\_name jump.magedu.com;

index index.html index.php;

root /www/htdocs;

rewrite ^/ http://www.magedu.com/;

}

实现域名镜像

server

{

listen 80;

server\_name mirror.magedu.com;

index index.html index.php;

root /www/htdocs;

rewrite ^/(.\*)$ http://www.magedu.com/$1 last;

}

简单的防盗链配置：

location ~\* \.(gif|jpg|png|swf|flv)$ {

valid\_referers none blocked www.magedu.com;

if ($invalid\_referer) {

rewrite ^/ http://www.magedu.com/403.html;

# return 404

}

}

第一行：gif|jpg|png|swf|flv

表示对gif、jpg、png、swf、flv后缀的文件实行防盗链

第二行：www.magedu.com

表示对www.magedu.com这个来路进行判断if{}里面内容的意思是，如果来路不是指定来路就跳转到错误页面，当然直接返回404也是可以的。

if (!-e $request\_filename) {

rewrite ^/user/([0-9]+)/?$ /view.php?go=user\_$1 last;

rewrite ^/component/id/([0-9]+)/?$ /page.php?pageid=$1 last;

rewrite ^/component/([^/]+)/?$ /page.php?pagealias=$1 last;

rewrite ^/category\\_([0-9]+)\.htm$ http://$host/category/$1/ permanent;

rewrite ^/showday\\_([0-9]+)\\_([0-9]+)\\_([0-9]+)\.htm$ http://$host/date/$1/$2/$3/ permanent;

showday\_1\_2\_3.htm $host/date/1/2/3/

}

server {

listen 80 default;

server\_name \*.mysite.com;

rewrite ^ http://mysite.com$request\_uri permanent;

}

常用的变量：

$arg\_PARAMETER This variable contains the value of the GET request variable PARAMETER if present in the query string.

$args This variable contains the query string in the URL, for example foo=123&bar=blahblah if the URL is http://example1. com/? foo=123&bar=blahblah

$binary\_remote\_addr The address of the client in binary form.

$body\_bytes\_sent The bytes of the body sent.

$content\_length This variable is equal to line Content-Length in the header of request.

$content\_type This variable is equal to line Content-Type in the header of request.

$document\_root This variable is equal to the value of directive root for the current request.

$document\_uri The same as $uri.

$host This variable contains the value of the 'Host' value in the request header, or the name of the server processing if the 'Host' value is not available.

$http\_HEADER The value of the HTTP header HEADER when converted to lowercase and with "dashes" converted to "underscores", for example, $http\_user\_agent, $http\_referer.

$is\_args Evaluates to "?" if $args is set, returns "" otherwise.

$request\_uri This variable is equal to the \*original\* request URI as received from the client including the args. It cannot be modified. Look at $uri for the post-rewrite/altered URI. Does not include host name. Example: "/foo/bar.php?arg=baz".

$scheme The HTTP scheme (that is http, https). Evaluated only on demand, for example: rewrite ^(.+)$ $scheme://example.com$1 redirect;

$server\_addr This variable contains the server address. It is advisable to indicate addresses correctly in the listen directive and use the bind parameter so that a system call is not made every time this variable is accessed.

$server\_name The name of the server.

$server\_port This variable is equal to the port of the server, to which the request arrived.

$server\_protocol This variable is equal to the protocol of request, usually this is HTTP/1.0 or HTTP/1.1.

$uri This variable is equal to current URI in the request (without arguments, those are in $args.) It can differ from $request\_uri which is what is sent by the browser. Examples of how it can be modified are internal redirects, or with the use of index. Does not include host name. Example: "/foo/bar.html"

HTTP Headers

HTTP\_USER\_AGENT

HTTP\_REFERER

HTTP\_COOKIE

HTTP\_FORWARDED

HTTP\_HOST

HTTP\_PROXY\_CONNECTION

HTTP\_ACCEPT

Enabling a log file cache：

http {

...

open\_log\_file\_cache max=1000 inactive=20s min\_uses=2 valid=1m;

..

Simple Load Balancing

http {

upstream myproject {

server 172.16.100.11:80 weight=3;

server 172.16.100.12:80;

}

server {

listen 80;

server\_name www.a.com;

location / {

proxy\_pass http://myproject;

}

}

}

location / {

proxy\_pass http://10.1.1.2;

proxy\_set\_header X-Real-IP $remote\_addr;

}

upstream backend {

server backend1.example.com weight=5;

server backend2.example.com:8080;

}

server {

location / {

proxy\_pass http://backend;

}

}

upstream myhttpd {

server 10.1.1.2:80;

server 10.1.1.2:8080;

}

server {

location / {

proxy\_pass http://myhttpd;

}

location /nginx\_status {

# copied from http://blog.kovyrin.net/2006/04/29/monitoring-nginx-with-rrdtool/

stub\_status on;

access\_log off;

allow SOME.IP.ADD.RESS;

deny all;

}

active connections -- number of all open connections including connections to backends

server accepts handled requests -- nginx accepted 16630948 connections, handled 16630948 connections (no one was closed just it was accepted), and handles 31070465 requests (1.8 requests per connection)

reading -- nginx reads request header

writing -- nginx reads request body, processes request, or writes response to a client

waiting -- keep-alive connections, actually it is active - (reading + writing)

memcached

Memcached is an in-memory key-value store for small chunks of arbitrary data (strings, objects) from results of database calls, API calls, or page rendering.

Memcached is simple yet powerful. Its simple design promotes quick deployment, ease of development, and solves many problems facing large data caches.

http {

proxy\_cache\_path /var/www/cache levels=1:2 keys\_zone=mycache:20m

max\_size=2048m inactive=60m;

proxy\_temp\_path /var/www/cache/tmp;

...

server {

listen 80;

server\_name magedu.com;

access\_log /var/log/magedu.com/log/access.log;

error\_log /var/log/magedu.com/log/error.log debug;

#set your default location

location / {

proxy\_pass http://172.16.100.6/;

proxy\_cache mycache;

proxy\_cache\_valid 200 302 60m;

proxy\_cache\_valid 404 1m;

}

}

}

配置：

1、设定错误日志格式及级别：

http {

log\_format combined '$remote\_addr - $remote\_user [$time\_local] '

'"$request" $status $body\_bytes\_sent '

'"$http\_referer" "$http\_user\_agent"';

access\_log /var/log/nginx/access.log combined;

error\_log /var/log/nginx/error.log crit;

...

}

2、记录类似apache格式的日志：

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 /var/log/nginx/access.log main;

3、启用日志缓存：

http {

...

open\_log\_file\_cache max=1000 inactive=20s min\_uses=2 valid=1m;

...

}

1,1

Max Maximal number of descriptors in the cache, with overflow Least Recently Used removed (LRU)

Inactive Sets the time after which descriptor without hits during this time are removed; default is 10 seconds

min\_uses Sets the minimum number of file usage within the time specified in parameter inactive, after which the file descriptor will be put in the cache; default is 1

Valid Sets the time until it will be checked if file still exists under same name; default is 60 seconds

Off Disables the cache

设定限速：

1、为某个特定路径限速：

server {

server\_name www.magedu.com;

location /downloads/ {

limit\_rate 20k;

root /web/downloads/;

}

..

}

2、限制搜索引擎的bot速度：

if ($http\_user\_agent ~ Google|Yahoo|MSN|baidu) {

limit\_rate 20k;

}

The X-Forwarded-For (XFF) HTTP header field is a de facto standard for identifying the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer. This is an HTTP request header which was introduced by the Squid caching proxy server's developers. An effort has been started at IETF for standardizing the Forwarded-For HTTP header.

设定反向代理：

简单示例：

location / {

proxy\_pass http://www.internal.com:8080;

proxy\_set\_header X-Real-IP $remote\_addr;

}

X-Real-IP: 192.168.0.1

www.magedu.com

http://www.magedu.com

proxy\_pass

语法：proxy\_pass URL;

This directive sets the address of the proxied server and the URI to which location will be mapped. Address may be given as hostname or address and port. 例如：

proxy\_pass http://localhost:8000/uri/;

By default, the Host header from the request is not forwarded, but is set based on the proxy\_pass statement. To forward the requested Host header, it is necessary to use:

proxy\_set\_header Host $host;

proxy\_read\_timeout

语法：proxy\_read\_timeout time;

This directive sets the read timeout for the response of the proxied server. It determines how long nginx will wait to get the response to a request. The timeout is established not for entire response, but only between two operations of reading.

proxy\_send\_timeout

语法：roxy\_send\_timeout time;

This directive assigns timeout with the transfer of request to the upstream server. Timeout is established not on entire transfer of request, but only between two write operations. If after this time the upstream server will not take new data, then nginx is shutdown the connection.

nginx和后端http服务器之间的连接是通过http/1.0协议进行的，因此，每连接是单独建立的；但Nginx和客户端的browser之间的会话是基于http/1.1，因此可以实现keep-alive的功能。此外，在响应用户之前，nginx把每一个用户的会话缓存至本地。

其它常用指令：

proxy\_buffers

语法: proxy\_buffers the\_number is\_size;

This directive sets the number and the size of buffers, into which will be read the answer, obtained from the proxied server. By default, the size of one buffer is equal to the size of page.

例如：

proxy\_buffers 32 4k；

proxy\_set\_header

语法：proxy\_set\_header header value;

This directive allows to redefine and to add some request header lines which will be transferred to the proxied server.

例如：

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_connect\_timeout

语法：proxy\_connect\_timeout time;

This directive assigns a timeout for the connection to the upstream server. It is necessary to keep in mind that this time out cannot be more than 75 seconds.

proxy\_no\_cache

语法：proxy\_no\_cache variable1 variable2 ...;

定义不进行缓存的情形，例如：

proxy\_no\_cache $cookie\_nocache $arg\_nocache $arg\_comment;

proxy\_no\_cache $http\_pragma $http\_authorization;

为反向代理启用缓存功能：

http {

proxy\_cache\_path /data/nginx/cache levels=1:2 keys\_zone=STATIC:10m

inactive=24h max\_size=1g;

server {

location / {

proxy\_pass http://1.2.3.4;

proxy\_set\_header Host $host;

proxy\_cache STATIC;

proxy\_cache\_valid 200 1d;

proxy\_cache\_use\_stale error timeout invalid\_header updating

http\_500 http\_502 http\_503 http\_504;

}

}

}

proxy\_cache\_path

语法：proxy\_cache\_path path [levels=number] keys\_zone=zone\_name:zone\_size [inactive=time] [max\_size=size];

This directive sets the cache path and other cache parameters. Cached data is stored in files. An MD5 hash of the proxied URL is used as the key for the cache entry, and is also used as the filename in the cache path for the response contents and metadata.

The levels parameter sets the number of subdirectory levels in cache. You may use any combination of 1 and 2 in the level formats: X, X:X, or X:X:X e.g.: "2", "2:2", "1:1:2". There can be at most 3 levels.

All active keys and metadata is stored in shared memory. Zone name and the size of the zone is defined via the keys\_zone parameter.

If cached data is not requested for time defined by the inactive parameter, than that data is removed from the cache. The inactive parameter defaults to 10 minutes (10m).

proxy\_cache

语法：proxy\_cache zone\_name;

This directive sets name of zone for caching. The same zone can be used in multiple places.

The following response headers flag a response as uncacheable unless they are ignored:

Set-Cookie

Cache-Control containing "no-cache", "no-store", "private", or a "max-age" with a non-numeric or 0 value

Expires with a time in the past

X-Accel-Expires: 0

proxy\_cache\_valid

语法: proxy\_cache\_valid reply\_code [reply\_code ...] time;

设定对于不同类别应答的缓存时间. Example:

proxy\_cache\_valid 200 302 10m;

proxy\_cache\_valid 404 1m;

Also it is possible to cache any replies with parameter "any":

proxy\_cache\_valid 200 302 10m;

proxy\_cache\_valid 301 1h;

proxy\_cache\_valid any 1m;

反向代理多台服务器实现负载均衡：

upstream backend {

server www1.magedu.com weight=5;

server www2.magedu.com max\_fails=3 fail\_timeout=30s;

server www3.magedu.com;

}

server {

listen 80;

server\_name example1.com;

access\_log /var/log/magedu.com/access.log;

error\_log /var/log/magedu.com/error.log debug;

#set your default location

location / {

include proxy.conf;

proxy\_pass http://backend;

}

}

172.16.0.1

127.0.0.1:8080

server

语法：server name [parameters]

其中的name可以是FQDN，主机地址，端口或unix套接字；如果FQDN解析的结果为多个地址，则每个地址都会被用到；

weight = NUMBER - 设定权重，默认为1.

max\_fails = NUMBER - 在fail\_timeout指令设定的时间内发往此server的不成功的请求次数，达到此数目后，此服务器将变为不可操作状态；默认值为1；设定为0值则禁用此功能；

fail\_timeout = TIME - 默认为10秒；

down - marks server as permanently offline, to be used with the directive ip\_hash.

backup - (0.6.7 or later) only uses this server if the non-backup servers are all down or busy (cannot be used with the directive ip\_hash)

upstream

语法：upstream name { ... }

声明一组可以被proxy\_pass和fastcgi\_pass引用的服务器；这些服务器可以使用不同的端口，并且也可以使用Unix Socket；也可以为服务器指定不同的权重；例如：

upstream backend {

server backend1.magedu.com weight=5 down backup;

server 127.0.0.1:8080 max\_fails=3 fail\_timeout=30s;

server unix:/tmp/backend3;

}

LNMMP =

安装配置第三方模块，实现upstream中对后端http server的健康状态检测：

模块下载地址：https://github.com/cep21/healthcheck\_nginx\_upstreams；模块名称：ngx\_http\_healthcheck\_module

安装配置方法：

1、首先解压healcheck模块到某路径下，这里假设为/tmp/healthcheck\_nginx\_upstreams

2、对nginx打补丁

首先解压nginx，并进入nginx源码目录：

# tar xf nginx-1.0.11.tar.gz

# cd nginx-1.0.11

# patch -p1 < /tmp/healthcheck\_nginx\_upstreams/nginx.patch

而后编译nginx，在执行configure时添加类似下面的选项：

--add-module=/tmp/healthcheck\_nginx\_upstreams

所以，这里就使用如下命令：

# ./configure \

--prefix=/usr \

--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/nginx.pid \

--lock-path=/var/lock/nginx.lock \

--user=nginx \

--group=nginx \

--with-http\_ssl\_module \

--with-http\_flv\_module \

--with-http\_stub\_status\_module \

--with-http\_gzip\_static\_module \

--http-client-body-temp-path=/var/tmp/nginx/client/ \

--http-proxy-temp-path=/var/tmp/nginx/proxy/ \

--http-fastcgi-temp-path=/var/tmp/nginx/fcgi/ \

--with-pcre \

--add-module=/tmp/healthcheck\_nginx\_upstreams

# make && make install

ngx\_http\_healthcheck\_module模块的使用方法：

1、此模块支持的指令有：

healthcheck\_enabled

启用此模块

healthcheck\_delay

对同一台后端服务器两次检测之间的时间间隔，单位毫秒，默认为1000；

healthcheck\_timeout

进行一次健康检测的超时时间，单位为毫秒，默认值2000；

healthcheck\_failcount

对一台后端服务器检测成功或失败多少次之后方才确定其为成功或失败，并实现启用或禁用此服务器；

healthcheck\_send

为了检测后端服务器的健康状态所发送的检测请求；如：healthcheck\_send "GET /health HTTP/1.0" 'Host: www.magedu.com';

healthcheck\_expected

期望从后端服务器收到的响应内容；如果未设置，则表示从后端服务器收到200状态码即为正确；

healthcheck\_buffer

健康状态检查所使用的buffer空间大小；

healthcheck\_status

通过类似stub\_status的方式输出检测信息，使用方法如下：

location /stat {

healthcheck\_status;

}

一个例子：

http {

upstream backend {

server 127.0.0.1:8080;

server 172.16.0.1:80;

healthcheck\_enabled;

healthcheck\_delay 1000;

healthcheck\_timeout 1000;

healthcheck\_failcount 3;

healthcheck\_expected 'I\_AM\_ALIVE';

healthcheck\_send "GET /.health HTTP/1.0";

# Optional supervisord module support

#supervisord none;

#supervisord\_inherit\_backend\_status;

}

server {

listen 80;

location / {

proxy\_set\_header Host $http\_host;

proxy\_pass http://backend;

proxy\_connect\_timeout 3;

}

location /stat {

healthcheck\_status;

}

}

}

Nginx整合memcached:

server {

listen 80;

server\_name www.magedu.com;

#charset koi8-r;

#access\_log logs/host.access.log main;

location / {

set $memcached\_key $uri;

memcached\_pass 127.0.0.1:11211;

default\_type text/html;

error\_page 404 @fallback;

}

location @fallback {

proxy\_pass http://172.16.0.1;

}

}