第二十七章：apache应用详解及优化

**一、apache优化之保持连接、网页传输压缩、页面缓存时间；**

**二、apache优化隐藏版本号；**

**三、apache优化之防盗链；**

**四、apache的ab压力测试；**

**五、apache 日志分析工具；**

**六、apache访问控制；**

**七、apache优化之日志分割；**

**八、apache实现虚拟主机；**

**案例环境：**

|  |  |  |  |
| --- | --- | --- | --- |
| 系统类型 | IP地址 | 主机名 | 所需软件 |
| Centos 6.5 64bit | 192.168.100.150 | www.linuxfan.cn | httpd-2.2.17.tar.gz |
| 注：  yum -y install zlib-devel ##安装--enable-deflate选项需要的网页压缩传输的软件包  ./configure --prefix=/usr/local/httpd --enable-cgi --enable-rewrite --enable-so --enable-deflate --enable-expires ##编译安装时的命令 | | | |

**一、apache优化之保持连接、网页传输压缩、页面缓存时间；**

**1.保持连接、长链接（keep-alive）：**

作用：HTTP协议是TCP的子协议，HTTP建立连接时需要先建立TCP连接，断开时也需要端口TCP连接，TCP的建立和断开消耗的资源通过HTTP 连接保持可以节约下来。对于 HTTP/1.1，尽量地保持客户端的TCP连接，通过一个TCP连接传送多个 HTTP 请求响应，对于客户端可以提高 50%以上的响应时间，对于服务器可以降低建立与关闭TCP连接时的资源消耗。

**配置项：**

|  |  |
| --- | --- |
| keepalive ON|OFF | 是否打开保持连接功能。根据网站的并发请求量决定是否打开，高并发时打开，并发量不高时可选择关闭。 |
| keepalive timeout | 一次TCP连接多次HTTP请求之间的最大间隔时间，两次HTTP请求超过此时间连接将会断开。 |
| maxkeepaliverequestes | 一次TCP连接能够传输的最大HTTP请求数量。 |

[root@www ~]# vi /usr/local/httpd/conf/extra/httpd-default.conf

16 KeepAlive On

23 MaxKeepAliveRequests 100

29 KeepAliveTimeout 20

:wq

[root@www ~]# sed -n '/httpd-default.conf/p' /usr/local/httpd/conf/httpd.conf ##查看

#Include conf/extra/httpd-default.conf

[root@www ~]# sed -i '/httpd-default.conf/s/^#//g' /usr/local/httpd/conf/httpd.conf ##去除注释

[root@www ~]# sed -n '/httpd-default.conf/p' /usr/local/httpd/conf/httpd.conf ##查看验证

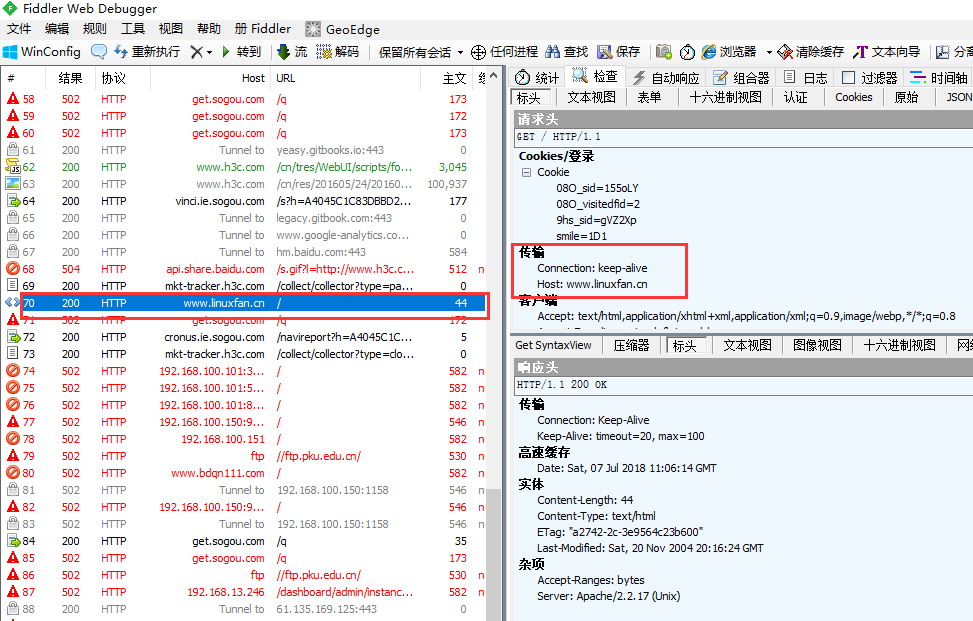
Include conf/extra/httpd-default.conf

[root@www ~]# sed -i '97s/^#//g' /usr/local/httpd/conf/httpd.conf

[root@www ~]# sed -i 's/www.example.com/www.linuxfan.cn/g' /usr/local/httpd/conf/httpd.conf

[root@www ~]# /usr/local/httpd/bin/apachectl start

**访问抓包验证：**



**2.apache网页压缩：**

**作用：**配置 Apache 的网页压缩功能，是使用 Gzip 压缩算法来对 Apache 服务器发布的网页内容进行压缩后再传输到客户端浏览器。通常在cpu有空闲，网站带宽占用高时，会启用压缩；

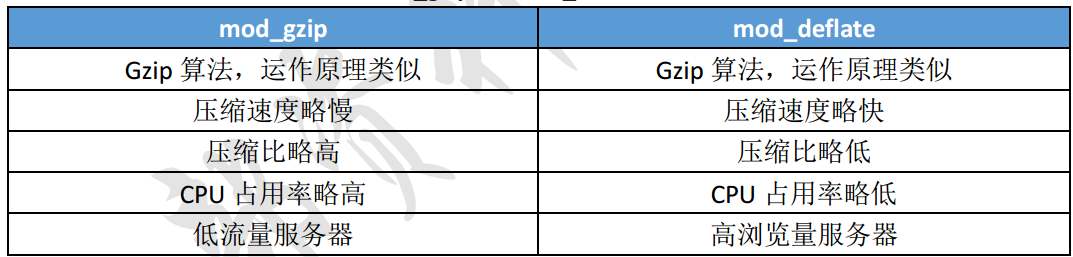
**优势：**

1）加快网页加载的速度，改善用户的浏览体验

2）降低网络传输带宽， 服务器节省流量

3）网页压缩有利于搜索引擎的抓取

**方式：**Apache 能实现网页压缩功能的模块有 mod\_gzip 模块和 mod\_deflate 模块



**注意：编译安装时需要使用以下配置项；**

--enable-deflate ##启用deflate模块，注意必须安装依赖包“zlib-devel”

**配置项：**

|  |  |
| --- | --- |
| AddOutputFilterByType DEFLATE text/html text/plain text/css text/xml text/javascript | 对什么格式的内容启用压缩 |
| DeflateCompressionLevel 9 | 压缩级别为9，范围是1-9，数字大压缩率高 |
| SetOutputFilter DEFLATE | 启用deflate压缩方式 |

[root@www ~]# /usr/local/httpd/bin/apachectl -D DUMP\_MODULES |grep deflate

deflate\_module (static)

Syntax OK

[root@www ~]# vi /usr/local/httpd/conf/httpd.conf ##在最后添加如下配置

AddOutputFilterByType DEFLATE text/html text/plain text/css text/xml text/javascript

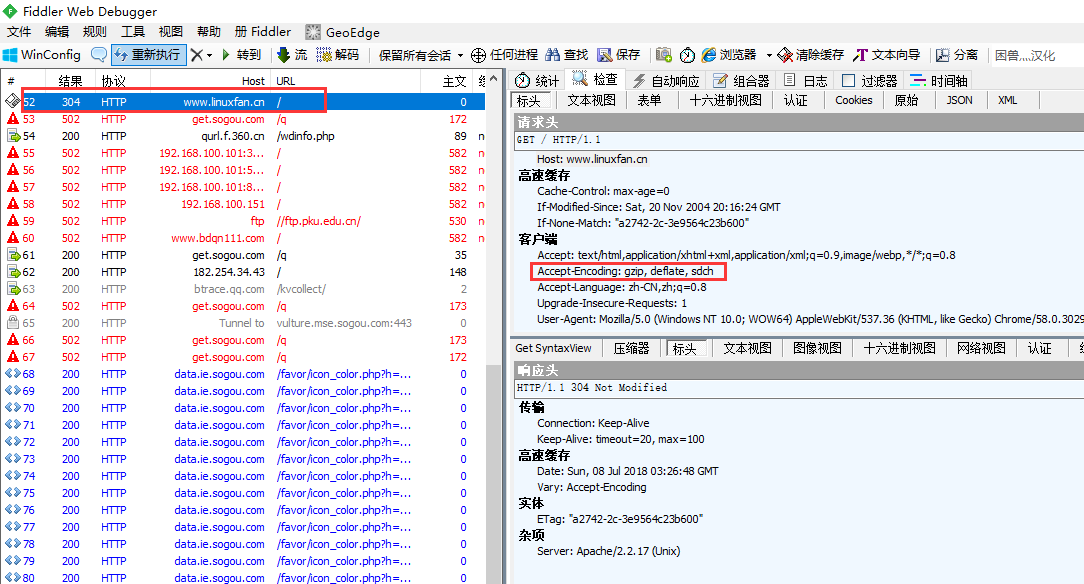
DeflateCompressionLevel 9

SetOutputFilter DEFLATE

:wq

[root@www ~]# /usr/local/httpd/bin/apachectl start

**访问并抓包验证：**



**3.apache页面缓存时间：**

**作用：**通过 mod\_expires 模块配置 Apache，使网页能在客户端浏览器缓存一段时间，以避免重复请求，减轻服务端工作压力。启用 mod\_expires 模块后，会自动生成页面头部信息中的 Expires 标签和 CacheControl 标签，从而降低客户端的访问频率和次数，达到减少不必要的流量和增加访问速度的目的。

**注意：编译安装时必须使用以下配置项**

--enable-expires ##启用网页缓存过期时间模块

**配置项：**

<IfModule mod\_expires.c>

ExpiresActive On ##启用页面缓存时间

ExpiresDefault "access plus 60 seconds" ##设置默认缓存1分钟

</IfModule>

[root@www ~]# /usr/local/httpd/bin/apachectl -D DUMP\_MODULES |grep expires

expires\_module (static)

Syntax OK

[root@www ~]# vi /usr/local/httpd/conf/httpd.conf ##在最后添加如下配置

<IfModule mod\_expires.c>

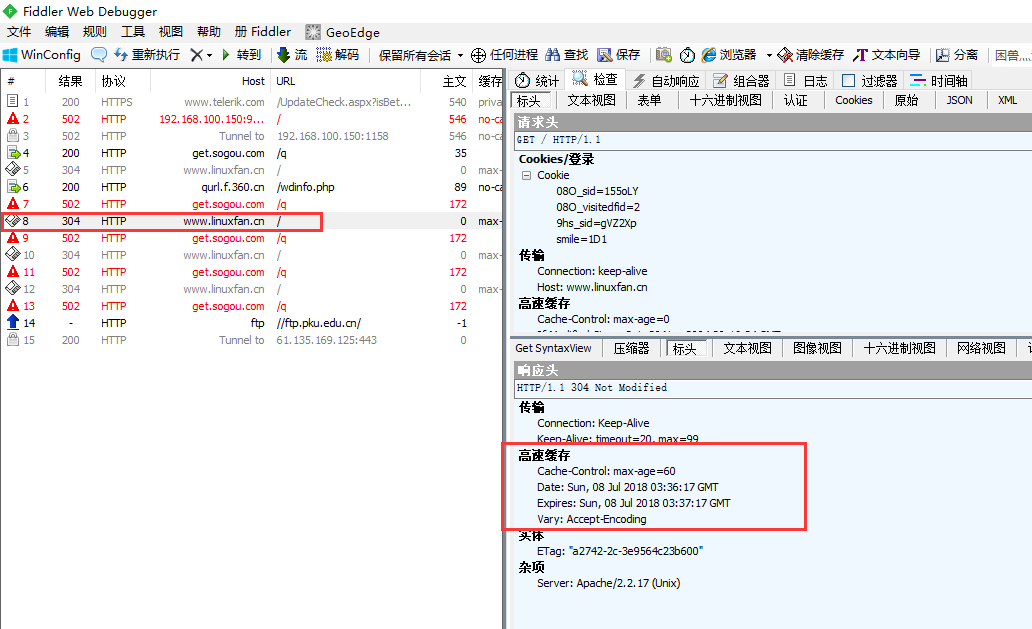
ExpiresActive On

ExpiresDefault "access plus 60 seconds"

</IfModule>

[root@www ~]# /usr/local/httpd/bin/apachectl restart

**清除浏览器历史记录，重新访问测试：**



**二、apache优化隐藏版本号；**

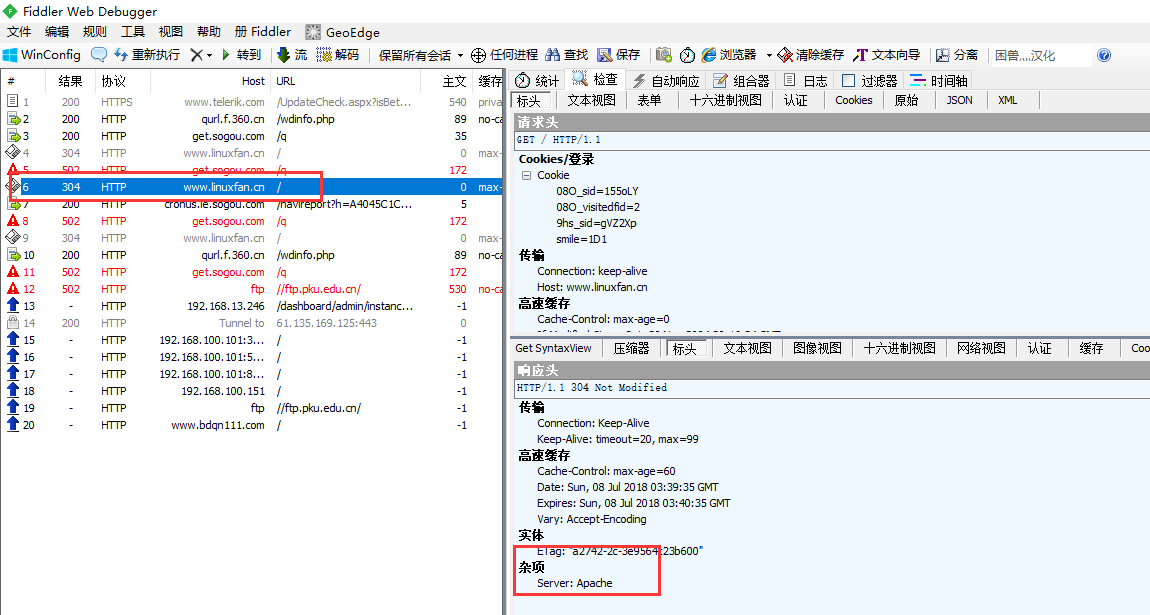
**方式一：**

[root@www ~]# vi /usr/local/httpd/conf/httpd.conf ##添加在末尾

ServerTokens Prod ##显示最少的信息，默认是Full显示完整信息

ServerSignature Off ##生成页面的页脚

**访问测试验证：**



**方式二：**

如果是源代码编译安装，还可以用修改源码编译的方法：进入 Apache 的源码目录下的 include 目录， 编辑 ap\_release.h 这个文件如下变量，然后重新编译安装即可：

42 #define AP\_SERVER\_BASEVENDOR "linuxfan.cn Software Foundation"

43 #define AP\_SERVER\_BASEPROJECT "Miscrosoft-IIS"

44 #define AP\_SERVER\_BASEPRODUCT "Microsoft-IIS"

45

46 #define AP\_SERVER\_MAJORVERSION\_NUMBER 6

47 #define AP\_SERVER\_MINORVERSION\_NUMBER 6

48 #define AP\_SERVER\_PATCHLEVEL\_NUMBER 6

**三、apache优化之防盗链；**

**作用：**防盗链就是防止别人盗用服务器中的图片、文件、视频等相关资源。运维人员可以通过apache提供rewrite模块进行优化。

**配置项：**

|  |  |
| --- | --- |
| RewriteEngine ON | ##打开网页重写功能 |
| RewriteCond | ##设置匹配规则 |
| RewriteRule | ##设置跳转动作 |

**rewrite的规则：**%{HTTP\_REFERER} 浏览 header 中的链接字段，存放一个连接的 URL，代表是从哪个链接访问所需的网页。

|  |  |
| --- | --- |
| !^ | 不以某个字段开头 |
| .\*$ | 以任意字符结尾 |
| NC | 不区分大小写 |
| R | 强制跳转 |

**规则匹配：**如果相应变量的值匹配所设置的规则，则逐条往下处理；如果不匹配，则后面的规则不再继续匹配。

**配置实例：**

[root@www ~]# vi /usr/local/httpd/conf/httpd.conf

在<Directory “/usr/local/httpd/htdocs”> 区域中，添加：

RewriteEngine On ##启用重写功能

RewriteCond %{HTTP\_REFERER} !^http://linuxfan.cn/.\*$ [NC]

RewriteCond %{HTTP\_REFERER} !^http://linuxfan.cn$ [NC]

RewriteCond %{HTTP\_REFERER} !^http://www.linuxfan.cn/.\*$ [NC]

RewriteCond %{HTTP\_REFERER} !^http://www.linuxfan.cn$ [NC]

RewriteRule .\*\.(gif|jpg|png|swf)$ http://www.linuxfan.cn/error.html [R,NC]

</Directoy>

最终匹配结果为：第二、三、四、五行的信任站点，能够使用网站上的图片；处理信任站点以外的站点，如果访问或使用 http://www.linuxfan.cn 域名以外的 gif、 jpg、 swf结尾的文件将跳转到重定向页面。

**案例：**

1.案例环境：

|  |  |  |  |
| --- | --- | --- | --- |
| 系统类型 | IP地址 | 主机名 | 所需软件 |
| Centos 6.5 64bit | 192.168.100.150 | linuxfan | httpd-2.2.17.tar.gz |
| Centos 6.5 64bit | 192.168.100.151 | linuxren | httpd-2.2.17.tar.gz |

2.安装配置linuxfan主机的httpd服务；

[root@linuxfan ~]# tar zxvf httpd-2.2.17.tar.gz -C /usr/src/

[root@linuxfan ~]# cd /usr/src/httpd-2.2.17/

[root@linuxfan httpd-2.2.17]#./configure --prefix=/usr/local/httpd --enable-cgi --enable-rewrite --enable-so && make && make install

[root@linuxfan httpd-2.2.17]# cd

[root@linuxfan ~]# sed -i '97s/^#//g' /usr/local/httpd/conf/httpd.conf

[root@linuxfan ~]# sed -i 's/www.example.com/www.linuxfan.cn/g' /usr/local/httpd/conf/httpd.conf

[root@linuxfan ~]# cat <<END >/usr/local/httpd/htdocs/index.html

<html>

<head><title></title></head>

<body>

<h1>www.linufan.cn</h1>

<img src="./linux.jpg" />

</body>

</html>

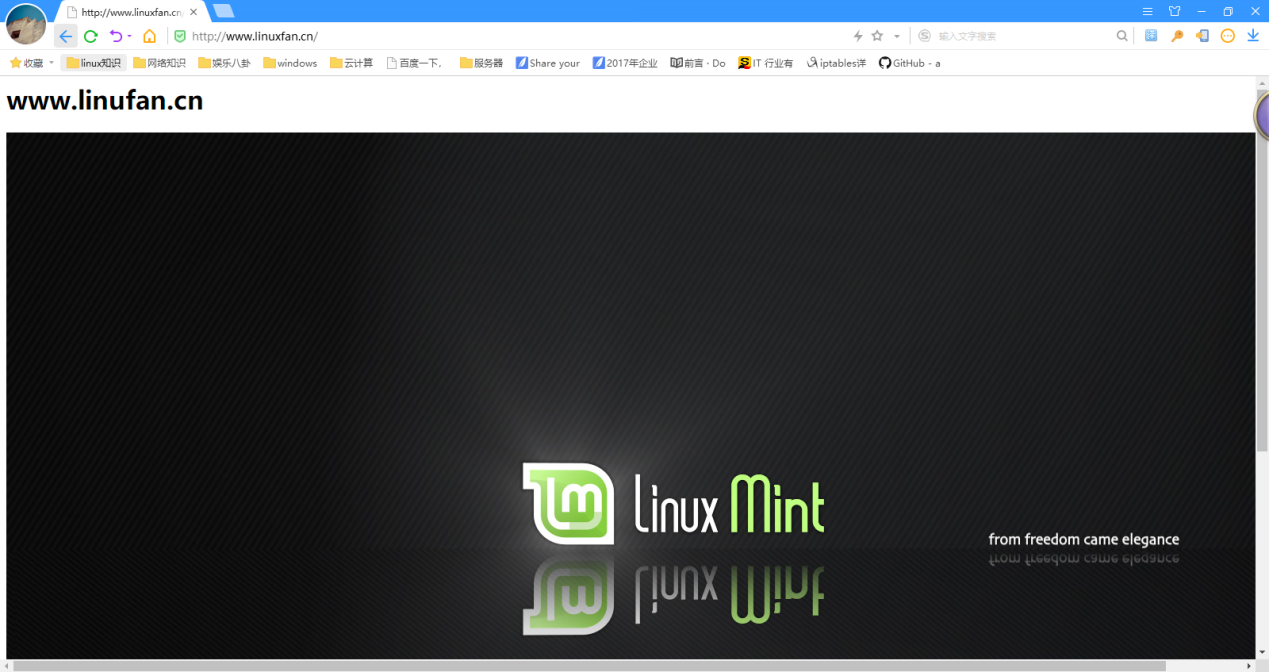
END

[root@linuxfan ~]# ls /usr/local/httpd/htdocs/

index.html linux.jpg

[root@linuxfan ~]# /usr/local/httpd/bin/apachectl start

访问测试：



3.安装配置linuxren主机的httpd服务；

[root@linuxren ~]# tar zxvf httpd-2.2.17.tar.gz -C /usr/src/

[root@linuxren ~]# cd /usr/src/httpd-2.2.17/

[root@linuxren httpd-2.2.17]#./configure --prefix=/usr/local/httpd --enable-cgi --enable-rewrite --enable-so && make && make install

[root@linuxren httpd-2.2.17]# cd

[root@linuxren ~]# sed -i '97s/^#//g' /usr/local/httpd/conf/httpd.conf

[root@linuxren ~]# sed -i 's/www.example.com/www.linuxren.cn/g' /usr/local/httpd/conf/httpd.conf

[root@linuxren ~]# cat <<END >/usr/local/httpd/htdocs/index.html

<html>

<head><title> </title></head>

<body>

<h1>www.linuxren.cn </h1>

<img src="http://www.linuxfan.cn/linux.jpg" alt="访问被拒绝了！对方设置了防盗链！" />

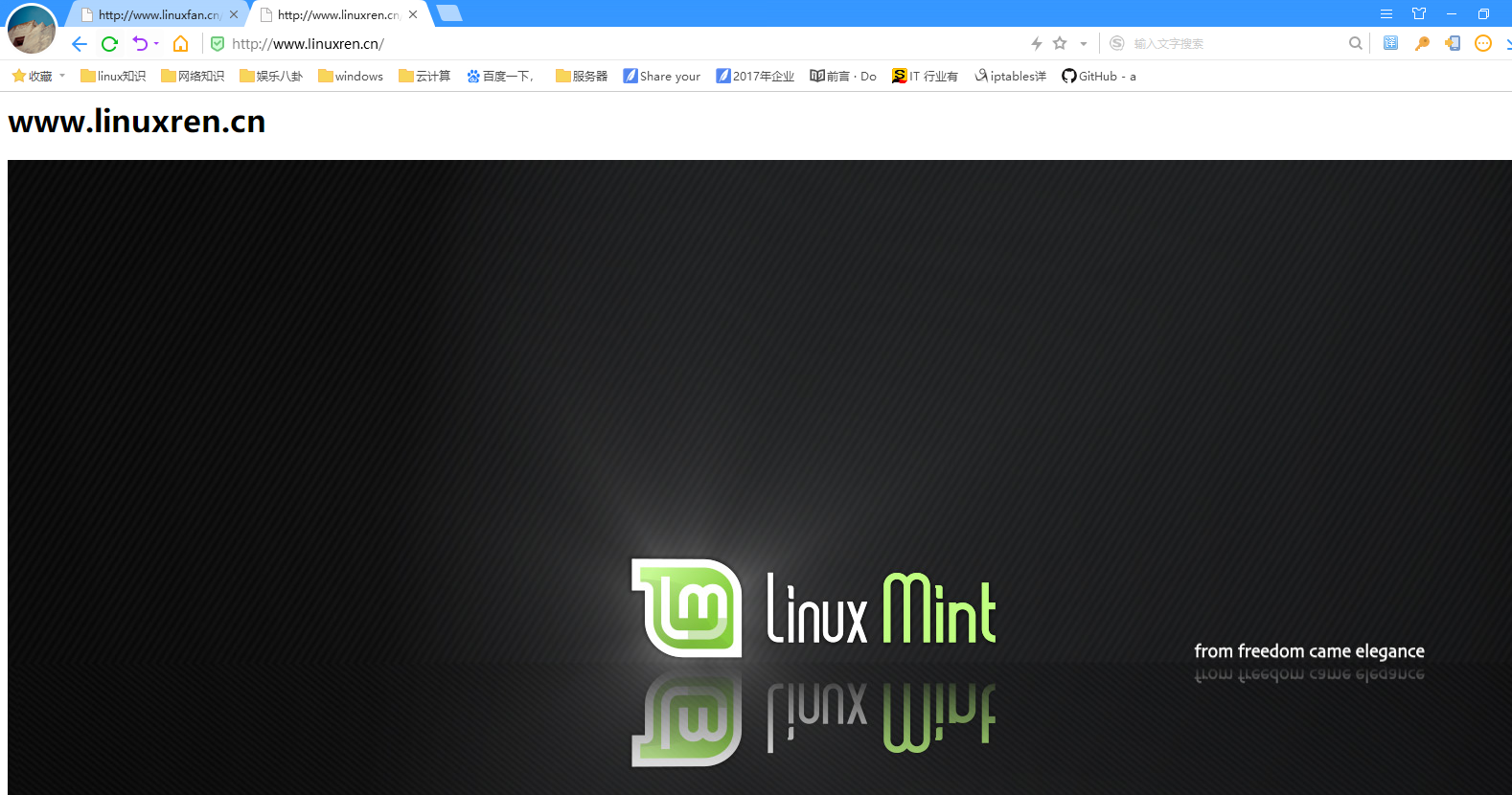
</body>

</html>

END

[root@linuxren ~]# /usr/local/httpd/bin/apachectl start

访问测试：



4.为linuxfan网站设置防盗链，防止linuxren网站盗用图片；

[root@linuxfan ~]# vi /usr/local/httpd/conf/httpd.conf

132 <Directory "/usr/local/httpd/htdocs">

133 RewriteEngine On

134 RewriteCond %{HTTP\_REFERER} !^http://linuxfan.cn/.\*$ [NC]

135 RewriteCond %{HTTP\_REFERER} !^http://linuxfan.cn$ [NC]

136 RewriteCond %{HTTP\_REFERER} !^http://www.linuxfan.cn/.\*$ [NC]

137 RewriteCond %{HTTP\_REFERER} !^http://www.linuxfan.cn$ [NC]

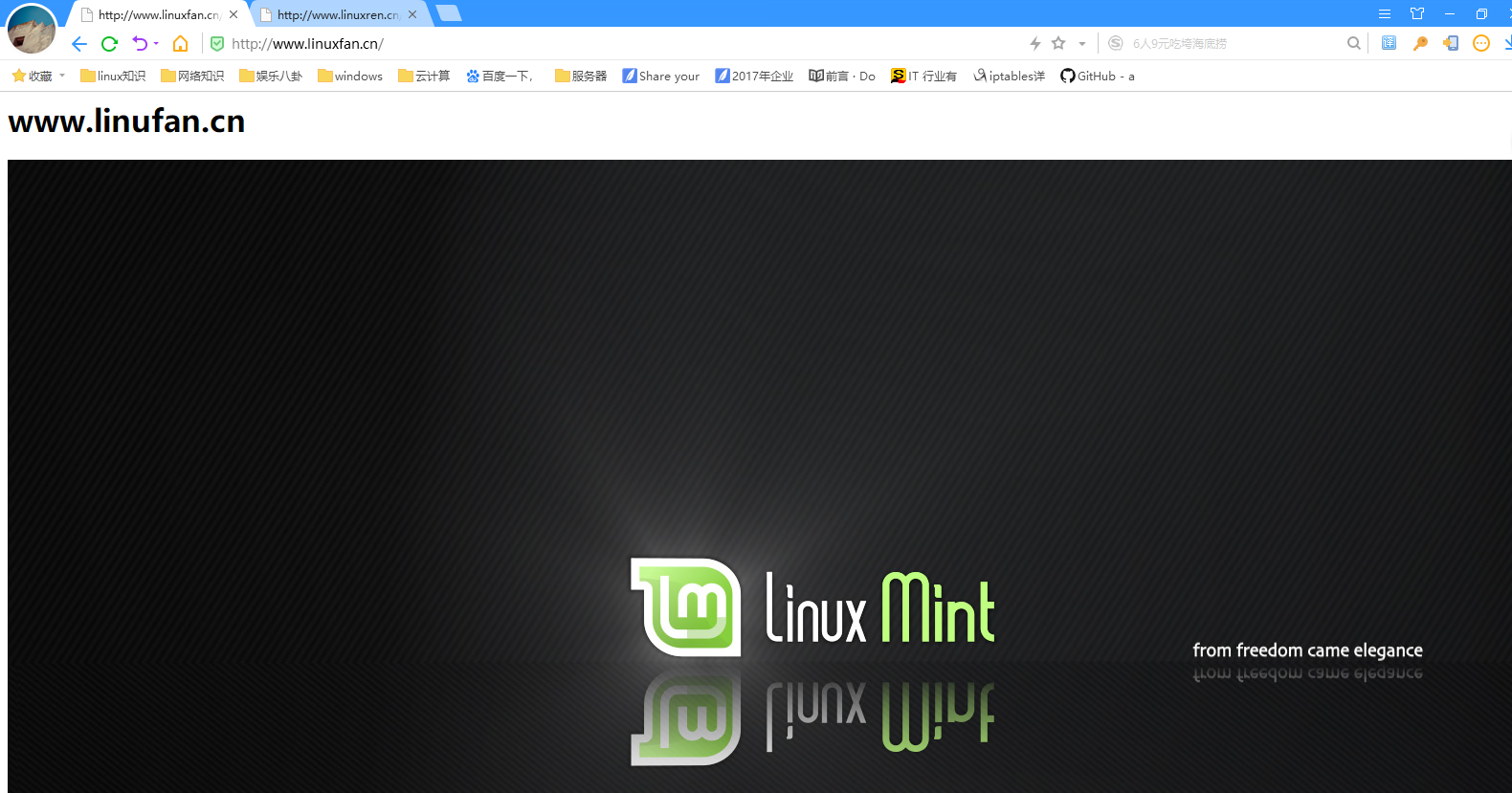
138 RewriteRule .\*\(gif|jpg|png|swf\)$ http://www.linuxfan.cn/error.html [R,NC]

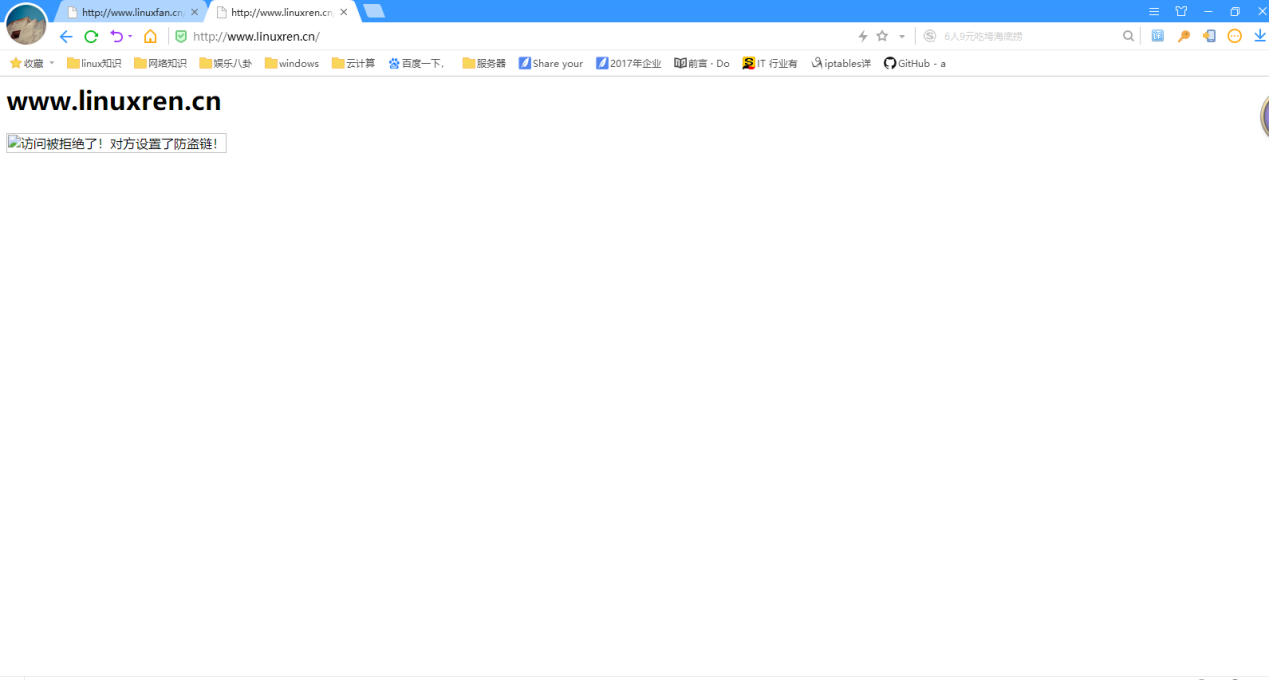
:wq

[root@linuxfan ~]# echo "deny" >>/usr/local/httpd/htdocs/error.html

[root@linuxfan ~]# /usr/local/httpd/bin/apachectl restart

访问验证：





**四、apache的ab压力测试；**

**作用：**Apache 附带了压力测试工具 ab，非常容易使用，并且完全可以模拟各种条件对 Web 服务器发起测试请求。在进行性能调整优化过程中，可用 ab 压力测试工具进行优化效果的测试。

**语法：**ab -n 请求数总和 -c 并发用户数 网站网址

**重要参数：**

Time taken for tests ##表示所有这些请求被处理完成所花费的总时间

Failed requests ##表示失败的请求数量

Requests per second ##吞吐率，计算公式： Complete requests / Time taken for tests

Transfer rate ##表示这些请求在单位时间内从服务器获取的数据长度，计算公式：Total trnasferred/ Time taken for tests，这个统计很好的说明服务器的处理能力达到极限时，其出口宽带的需求量。

[root@linuxfan ~]# /usr/local/httpd/bin/ab -n 1000 -c 80 http://192.168.100.150/index.html

**拓展：**自行百度apache其他性能压力测试工具

**五、apache 日志分析工具；**

**方式一：apachetop工具**

**作用：**查看服务器的实时运行情况，比如哪些 URL 的访问量最大，服务器每秒的请求数，等等。apachetop 就是这样一个工具， 能够让你实时的监测 apache 服务器的运行状况。

[root@linuxfan ~]# wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-6.repo

[root@linuxfan ~]# yum -y install epel-release

[root@linuxfan ~]# yum -y install apachetop

[root@linuxfan ~]# apachetop -f /usr/local/httpd/logs/access\_log

last hit: 06:14:54 atop runtime: 0 days, 00:00:05 06:14:55

All: 6 reqs ( 3.0/sec) 0.0B ( 0.0B/sec) 0.0B/req

2xx: 0 ( 0.0%) 3xx: 6 ( 100%) 4xx: 0 ( 0.0%) 5xx: 0 ( 0.0%)

R ( 5s): 6 reqs ( 1.2/sec) 0.0B ( 0.0B/sec) 0.0B/req

2xx: 0 ( 0.0%) 3xx: 6 ( 100%) 4xx: 0 ( 0.0%) 5xx: 0 ( 0.0%)

REQS REQ/S KB KB/S URL

3 1.50 0.0 0.0\*/

3 1.50 0.0 0.0 /linux.jpg

**方式二：awstats日志分析系统**

**作用：**Awstats 软件是一个由perl语言所编写的免费强大的服务器的日志文件分析工具，显示你所有的网页/邮件/ FTP统计包括访问，访问者，页面，点击，高峰时间，操作系统，浏览器，搜索引擎，关键字。

[root@linuxfan ~]# tar zxvf awstats-7.3.tar.gz

[root@linuxfan ~]# mv awstats-7.3 /usr/local/awstats

[root@linuxfan ~]# cd /usr/local/awstats/tools

[root@linuxfan tools]# chmod +x ./\*

[root@linuxfan tools]# ./awstats\_configure.pl

Do you want to continue setup from this NON standard directory [yN] ? Y

Config file path ('none' to skip web server setup):

> /usr/local/httpd/conf/httpd.conf

Do you want me to setup Apache to write 'combined' log files [y/N] ? Y

Do you want me to build a new AWStats config/profile

file (required if first install) [y/N] ? Y

Your web site, virtual server or profile name:

> www.linuxfan.cn

Directory path to store config file(s) (Enter for default): 回车

Press ENTER to continue... 回车

Press ENTER to finish... 回车

[root@linuxfan tools]# cd

[root@linuxfan ~]# vi /etc/awstats/awstats.www.linuxfan.cn.conf

LogFile="/usr/local/httpd/logs/access\_log" ##修改第 50 行

:wq

[root@linuxfan ~]# mkdir /var/lib/awstats

[root@linuxfan ~]# echo "\*/5 \* \* \* \* /usr/local/awstats/tools/awstats\_updateall.pl now" >/var/spool/cron/root

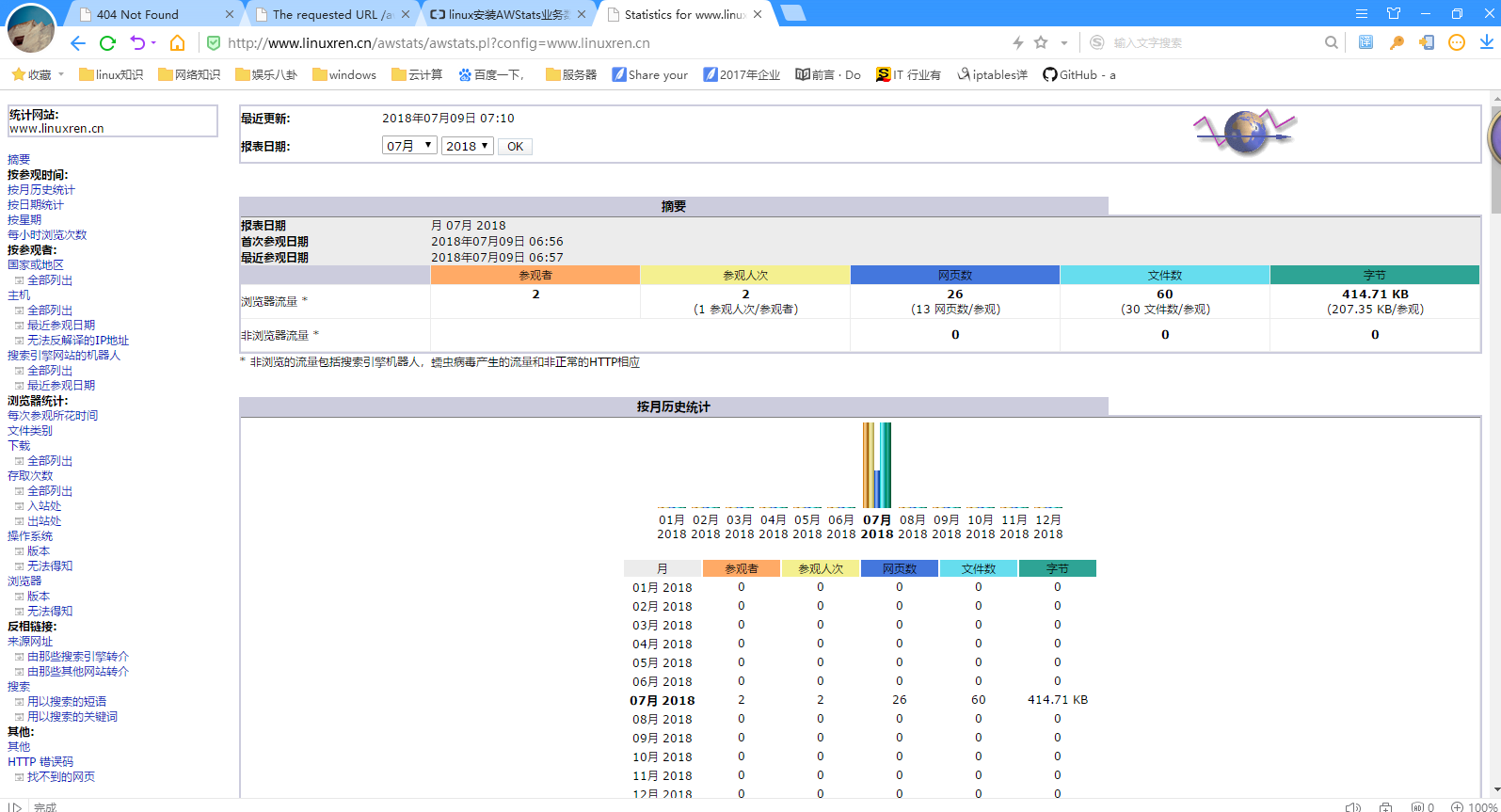
[root@linuxfan ~]# /etc/init.d/crond restart

[root@linuxfan ~]# /usr/local/awstats/tools/awstats\_updateall.pl now ##执行首次更新，如若更新访问数据无法更新，出现日志格式问题，需要调整日志格式为combined格式，如若已经为该模式，可将日志文件删除，重启服务生成即可

[root@linuxfan ~]# /usr/local/httpd/bin/apachectl restart

**访问测试：**

http://www.linuxfan.cn/awstats/awstats.pl?config=www.linuxfan.cn



**六、apache访问控制；**

**作用：**为apache服务提供的页面设置客户端访问权限，为某个组或者某个用户加密访问；

[root@linuxfan ~]# /usr/local/httpd/bin/htpasswd -c /usr/local/httpd/conf/htpasswd zs ##添加admin用户，可以在两个路径中间添加-c是新建文件删除原文件

New password: ##输入密码pwd@123

Re-type new password: ##确认密码

Adding password for user zs

[root@linuxfan ~]# cat /usr/local/httpd/conf/htpasswd

zs:TpPLVJuXl0wsE

[root@linuxfan ~]# /usr/local/httpd/bin/htpasswd /usr/local/httpd/conf/htpasswd ls

New password: ##输入密码pwd@123

Re-type new password: ##确认密码

Adding password for user ls

[root@linuxfan ~]# /usr/local/httpd/bin/htpasswd /usr/local/httpd/conf/htpasswd ww

New password: ##输入密码pwd@123

Re-type new password: ##确认密码

Adding password for user ww

[root@linuxfan ~]# vi /usr/local/httpd/conf/htgroups ##为授权用户加入组

mygroup: zs ls ##组名: 成员1 成员2

[root@linuxfan ~]# vim /usr/local/httpd/conf/httpd.conf

421 <Directory "/usr/local/awstats/wwwroot">

422 Options None

423 AllowOverride None

424 Order allow,deny

425 Allow from 192.168.100.1

426 AuthType Basic ##定义认证的类型为Basic

427 AuthName "Log analysis system" ##提示短语

428 AuthBasicProvider file ##提供认证者为file

429 AuthUserFile /usr/local/httpd/conf/htpasswd ##指定认证用户文件

430 AuthGroupFile /usr/local/httpd/conf/htgroups ##指定认证组文件

431 Require group mygroup ##设置允许访问的用户或者组，如果设置为所有用户改为“Require valid-user”即可，若设置单个用户为Require user 用户名 即可；

432 </Directory>

:wq

[root@linuxfan ~]# /usr/local/httpd/bin/apachectl restart

**访问测试：**

http://www.linuxfan.cn/awstats/awstats.pl?config=www.linuxfan.cn

##输入用户测试,zs与ls能登录，ww不能登录

**七、apache优化之日志分割；**

**作用：**随着网站访问量的增加，访问日志中的信息会越来越多， Apache 默认访问日志access\_log单个文件会越来越大，日志文件体积越大，信息都在一个文件中，查看及分析信息会及不方便。

**实现方式：**

1.利用 apache 自带的日志轮循程序 rotatelogs

ErrorLog "|rotatelogs 命令的绝对路径 -l logs/error\_%Y%m%d\_%H%M%S.log 秒数"

CustomLog "| rotatelogs 命令的绝对路径-l logs/access\_%Y%m%d\_%H%M%S.log 秒数"

combined

2.利用一个发展已经比较成熟的日志轮循工具 cronolog

ErrorLog "|cronolog 绝对路径 -l /usr/local/httpd/logs/error\_%Y%m%d%H%M%S.log "

CustomLog "|cronnolog绝对路径 -l /usr/local/httpd/logs/access\_%Y%m%d%H%M%S.log" combined

3.利用定时任务和 shell 脚本做日志切割

**方式一：** 利用 apache 自带的日志轮循程序 rotatelogs日志分割：

[root@linuxfan ~]# vi /usr/local/httpd/conf/httpd.conf ##10秒切割一次日志文件(切割日志时间频率可以自行改变，根据PV来决定，)，-l选项是小写的L表示local time（本地时间）

216 CustomLog "|/usr/local/httpd/bin/rotatelogs -l /usr/local/httpd/logs/access\_%Y%m%d%H%M%S.log 10" combined

:wq

[root@linuxfan ~]# /usr/local/httpd/bin/apachectl restart

[root@linuxfan ~]# ls /usr/local/httpd/logs/

access\_20171205054810.log error\_log

access\_20171205054820.log httpd.pid

**方式二：**利用一个发展已经比较成熟的日志轮循工具 cronolog

[root@linuxfan ~]# tar xf cronolog-1.6.2.tar.gz -C /usr/src/

[root@linuxfan ~]# cd /usr/src/cronolog-1.6.2/

[root@linuxfan cronolog-1.6.2]# ./configure && make && make install

[root@linuxfan cronolog-1.6.2]# cd

[root@linuxfan ~]# which cronolog

/usr/local/sbin/cronolog

[root@linuxfan ~]# vi /usr/local/httpd/conf/httpd.conf ##可以跟据文件名中的变量来设置分割的频率，Y表示四位年份、m表示月份、d表示月份中的天数

213 CustomLog "|/usr/sbin/cronolog /usr/local/httpd/logs/access\_%Y%m%d.log" combined

:wq

[root@linuxfan ~]# /usr/local/httpd/bin/apachectl restart

[root@linuxfan ~]# ls /usr/local/httpd/logs/

access\_20171206.log access\_log error\_log httpd.pid

**方式三：**脚本完成（定期移动httpd日志文件，并且保存期限为30天）：

[root@linuxfan ~]# vi /opt/cut\_httpd\_log.sh

#!/bin/bash

logpath="/usr/local/httpd/logs"

datetime=$(date -d "-1 day" "+%Y%m%d")

[ -d $logpath/backup ] || mkdir $logpath/backup

mv $logpath/access\_log $logpath/backup/access\_log-$datetime

/usr/local/httpd/bin/apachectl graceful

##重新加载apache的配置，不会中断请求

find $logpath/backup/ -mtime +30 | xargs rm –f

[root@linuxfan ~]# chmod +x /opt/cut\_httpd\_log.sh

[root@linuxfan ~]# echo "00 00 \* \* \* /bin/bash /opt/cut\_httpd\_log.sh &> /dev/null" >>/var/spool/cron/root

[root@linuxfan ~]# service crond restart

**日志合并：**

可以将每个服务器每天的日志文件通过 rsync 下载到专门进行访问统计分析的服务器上进行合并。合并多个服务器的日志文件，例如： log1 log2 log3 并输出到 log\_all 中的方法是：

sort -m -t " " -k 4 -o log\_all access\_log1 access\_log2 access\_log3

-m 使用 merge 优化算法

-t 指定排序的分割符

-k 4 指定排序时行中多列时，排序的依据列为哪列，表示根据时间进行排序

-o 表示将排序结果存放到指定的文件中

**八、apache实现虚拟主机；**

**作用：**在同一台服务器实现部署多个网站站点，节省资源；

**实现方式：**

1.不同IP，不同域名，相同端口；

2.相同IP，相同域名，不同端口；

3.相同IP，相同端口，不同域名；

**方式一：**不同IP，不同域名，相同端口；

[root@linuxfan ~]# ifconfig eth0:0 192.168.100.200

[root@linuxfan ~]# ip a |grep 192.168.100

inet 192.168.100.150/24 brd 192.168.100.255 scope global eth0

inet 192.168.100.200/24 brd 192.168.100.255 scope global secondary eth0:0

[root@linuxfan ~]# vi /usr/local/httpd/conf/httpd.conf

40 Listen 192.168.100.150:80

41 Listen 192.168.100.200:80

Include conf/extra/httpd-vhosts.conf ##去掉注释

[root@linuxfan ~]# vi /usr/local/httpd/conf/extra/httpd-vhosts.conf

NameVirtualHost 192.168.100.150:80

NameVirtualHost 192.168.100.200:80

<VirtualHost 192.168.100.150:80>

ServerAdmin admin@linuxfan.cn

DocumentRoot "/usr/local/httpd/htdocs/linuxfan/"

ServerName www.linuxfan.cn

ErrorLog "logs/www.linuxfan.cn-error\_log"

CustomLog "logs/www.linuxfan.cn-access\_log" common

</VirtualHost>

<VirtualHost 192.168.100.200:80>

ServerAdmin admin@linuxren.cn

DocumentRoot "/usr/local/httpd/htdocs/linuxren/"

ServerName www.linuxren.cn

ErrorLog "logs/www.linuxren.cn-error\_log"

CustomLog "logs/www.linuxren.cn-access\_log" common

</VirtualHost>

[root@linuxfan ~]# mkdir /usr/local/httpd/htdocs/linuxfan

[root@linuxfan ~]# mkdir /usr/local/httpd/htdocs/linuxren

[root@linuxfan ~]# echo "www.linuxfan.cn" >/usr/local/httpd/htdocs/linuxfan/index.html

[root@linuxfan ~]# echo "www.linuxren.cn" >/usr/local/httpd/htdocs/linuxren/index.html

[root@linuxfan ~]# /usr/local/httpd/bin/apachectl start

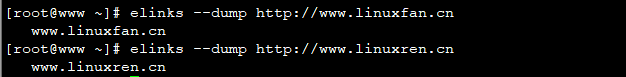
[root@linuxfan ~]# netstat -utpln |grep 80

tcp 0 0 192.168.100.200:80 0.0.0.0:\* LISTEN 53981/httpd

tcp 0 0 192.168.100.150:80 0.0.0.0:\* LISTEN 53981/httpd

访问测试：





**方式二：**相同IP，相同域名，不同端口；

[root@linuxfan ~]# vi /usr/local/httpd/conf/httpd.conf

Listen 192.168.100.150:80

Listen 192.168.100.150:8080

Include conf/extra/httpd-vhosts.conf ##去掉注释

[root@linuxfan ~]# vi /usr/local/httpd/conf/extra/httpd-vhosts.conf

NameVirtualHost 192.168.100.150:80

NameVirtualHost 192.168.100.150:8080

<VirtualHost 192.168.100.150:80>

ServerAdmin admin@linuxfan.cn

DocumentRoot "/usr/local/httpd/htdocs/linuxfan/"

ServerName www.linuxfan.cn

ErrorLog "logs/www.linuxfan.cn-error\_log"

CustomLog "logs/www.linuxfan.cn-access\_log" common

</VirtualHost>

<VirtualHost 192.168.100.150:8080>

ServerAdmin admin@linuxren.cn

DocumentRoot "/usr/local/httpd/htdocs/linuxren/"

ServerName www.linuxfan.cn

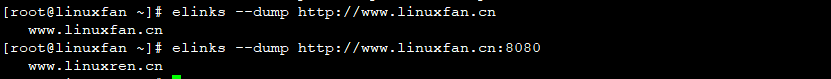
ErrorLog "logs/www.linuxren.cn-error\_log"

CustomLog "logs/www.linuxren.cn-access\_log" common

</VirtualHost>

[root@linuxfan ~]# /usr/local/httpd/bin/apachectl restart

访问测试：



**方式三：**相同IP，相同端口，不同域名；

[root@linuxfan ~]# vi /usr/local/httpd/conf/httpd.conf

Listen 192.168.100.150:80

Include conf/extra/httpd-vhosts.conf ##去掉注释

[root@linuxfan ~]# vi /usr/local/httpd/conf/extra/httpd-vhosts.conf

NameVirtualHost 192.168.100.150:80

<VirtualHost 192.168.100.150:80>

ServerAdmin admin@linuxfan.cn

DocumentRoot "/usr/local/httpd/htdocs/linuxfan/"

ServerName www.linuxfan.cn

ErrorLog "logs/www.linuxfan.cn-error\_log"

CustomLog "logs/www.linuxfan.cn-access\_log" common

</VirtualHost>

<VirtualHost 192.168.100.150:80>

ServerAdmin admin@linuxren.cn

DocumentRoot "/usr/local/httpd/htdocs/linuxren/"

ServerName www.linuxren.cn

ErrorLog "logs/www.linuxren.cn-error\_log"

CustomLog "logs/www.linuxren.cn-access\_log" common

</VirtualHost>

[root@linuxfan ~]# /usr/local/httpd/bin/apachectl restart

访问测试：

