第二十五章：Web基础与apache讲解

**一、网站基础；**

**二、HTTP协议分析；**

**三、TCP数据包状态分析；**

**四、apache原理详解；**

**五、案例：搭建apache网站服务；**

**一、网站基础：**

1.域名：域名是一个IP地址的“面具” ，目的是便于记忆和访问一个或一组服务器的地址（网站，电子邮件，FTP等）。

2.域名解析：本地HOSTS解析、DNS服务器解析

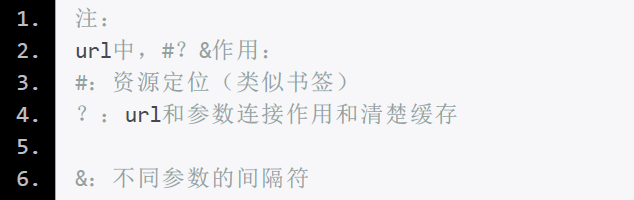
3.网站的基本概念：网站、网页、主页；HTTP、URL、HTML、超链接

4.web网站：有web1.0（以编辑为特征）和web2.0（侧重用户交互）

5.动态页面与静态页面的差别：

1）URL不同：静态页面没有?、动态页面包含？





2）后缀不同(开发语言不同)：静态页面一般以.html/.htm/.xml；动态页面一般以.php/.asp/.jsp/.pl/.py/.cgi等为后缀

3）静态页面的内容是固定的，动态页面的内容会因用户、浏览器、时间、地点等而发生变化。

http://（协议头）cn.bing.com（域名）/（URI资源：路径/文件名）：

https://jx.tmall.com/?ali\_trackid=2:mm\_28347190\_2425761\_27186547:1512434578\_208\_1779858597

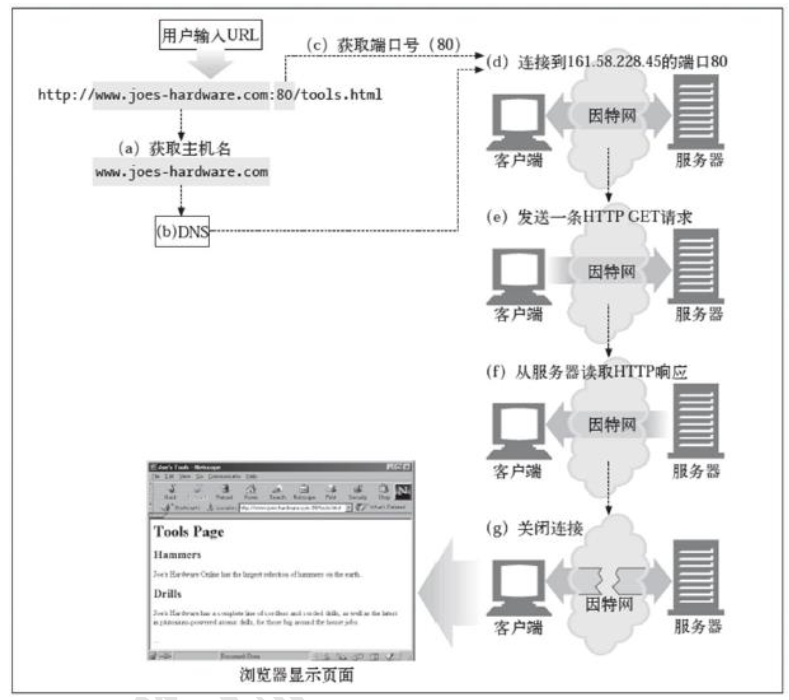
6.网站传输协议：http--超文本传输协议，监听的端口为tcp协议的80端口，不加密、https--安全超文本传输协议，监听的端口号是tcp协议的443号端口，加密传输，使用证书加密；

**二、HTTP协议分析：**

1.HTTP：超文本传输协议协议是互联网上应用最为广泛的一种网络协议，协议的核心功能是传输 Web 服务器上的 HTML 页面及其他文件；

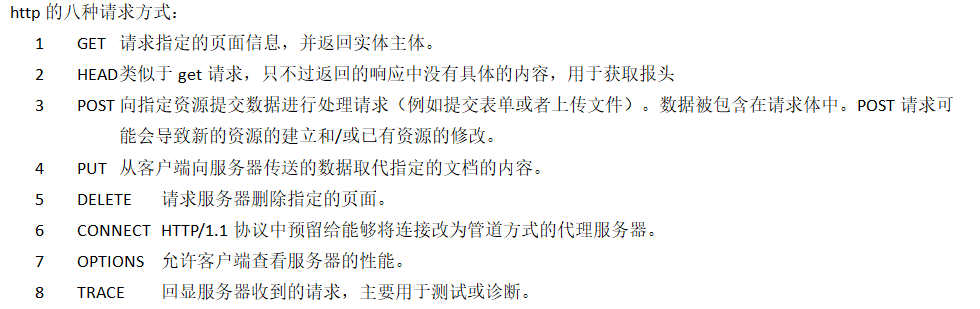
2.版本：http0.9（已过时）、http1.0和http1.1（目前广泛使用）、http2.0（未普及）；

3.http协议请求流程：



影响客户端访问web站点的因素：客户端请求的网络I/O；Web服务器请求页面的磁盘I/O；

4.HTTP 协议有多种获得 Web 资源的方法：



get方法与post方法的比较：

|  |  |  |
| --- | --- | --- |
|  | GET | POST |
| 对数据长度的限制 | URL最大的长度为2048字符 | 无限制 |
| 缓存 | 能被缓存 | 不能被缓存 |
| 安全性 | 较差 | 比较安全，参数不会出现在日志和浏览器的访问记录中 |
| 历史记录 | 保留在浏览器历史中 | 不保留 |

5.http状态返回状态码：

|  |  |  |
| --- | --- | --- |
| 状态码首位 | 范围 | 分类 |
| 1xx | 100-101 | 信息提示 |
| 2xx | 200-206 | 成功 |
| 3xx | 300-305 | 重定向 |
| 4xx | 400-415 | 客户端错误 |
| 5xx | 500-505 | 服务器错误 |

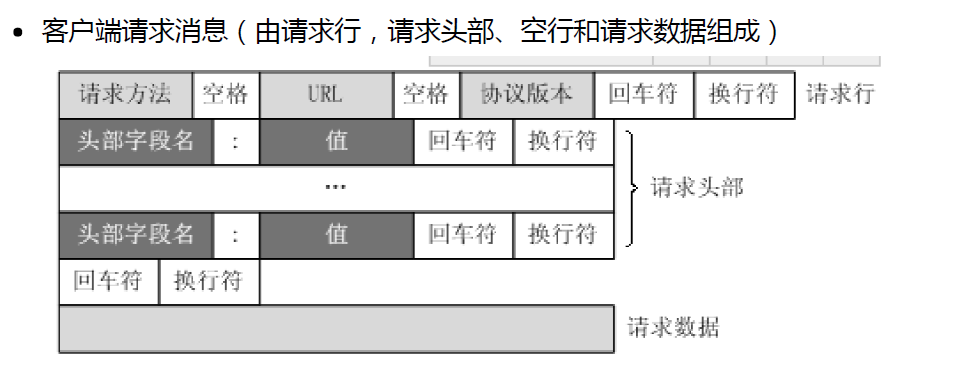
常见的状态返回码：

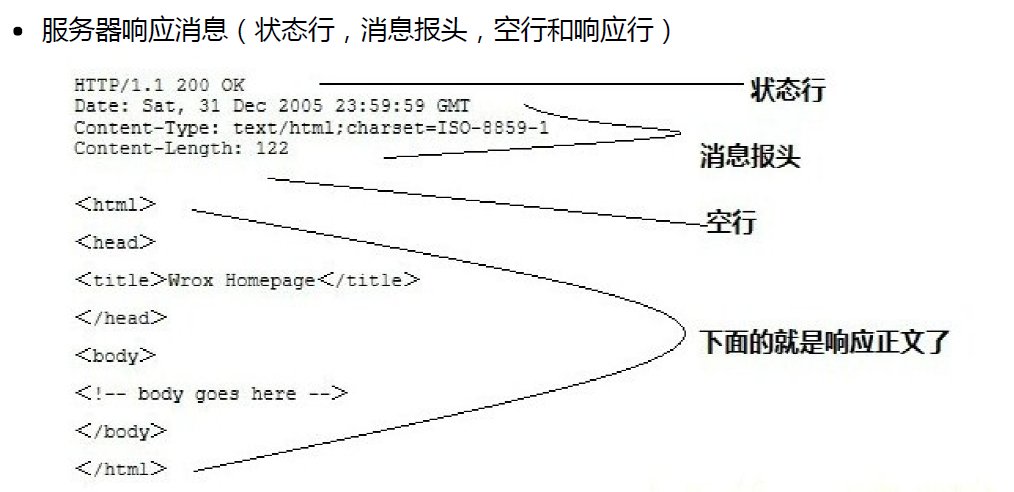
200 ok 301 请求的永久页面跳转 403 禁止访问该页面

404 找不到页面 500 服务器内部错误 502 网关错误

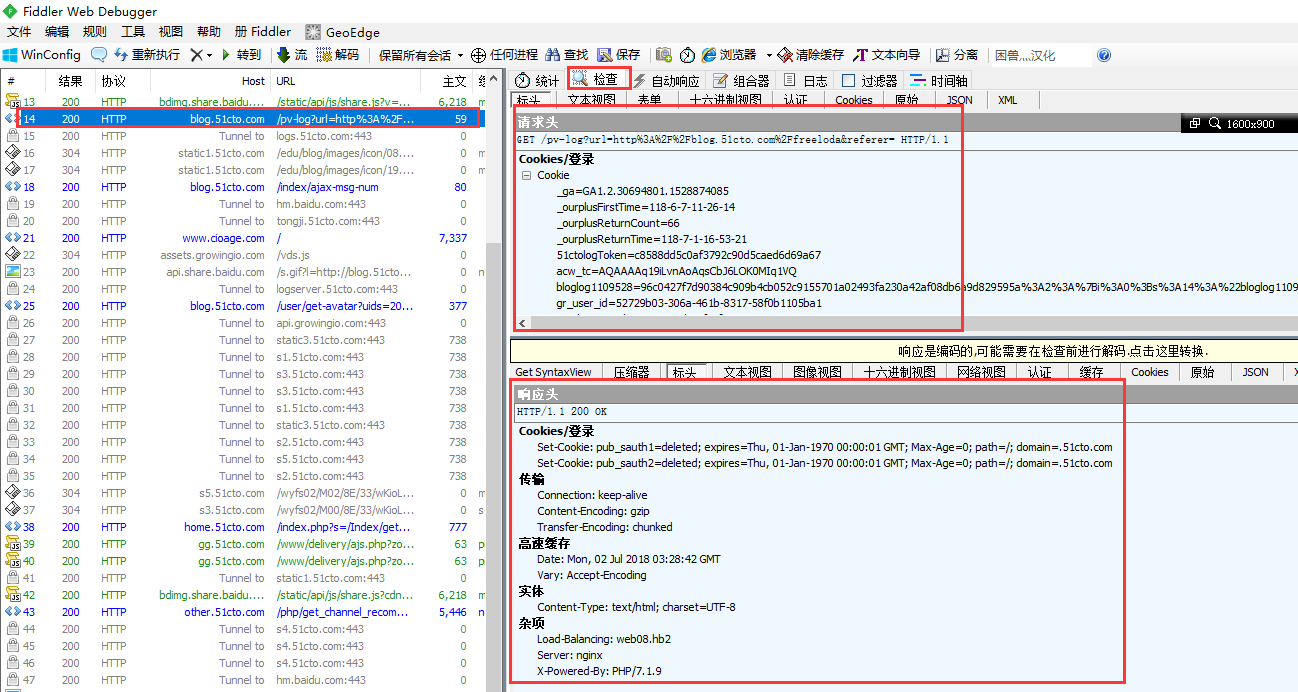
503 当前服务不可用 504 网关请求超时

6.http报文格式：一个完整的http访问包含请求（request）和响应（response）；



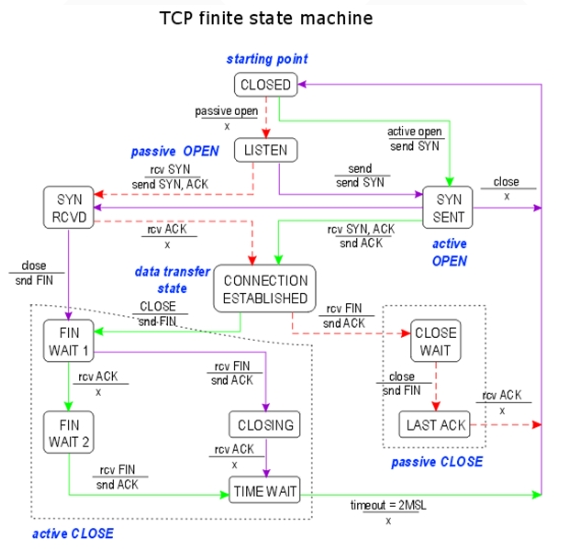


7.使用fiddler抓包工具抓包，查看http报文格式：

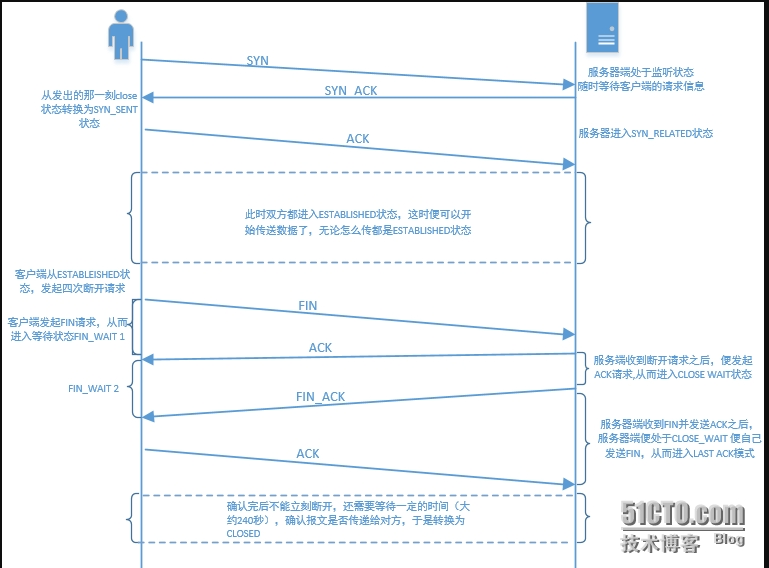


**三、TCP数据包状态分析：**

1.tcp报文走向：



2.TCP建立连接与断开连接时，连接的状态变化：



3.查看系统中数据包的状态；

netstat -anto ##查看系统中所有连接

netstat -ant |awk '/^tcp|^udp/{state[$6]++}END{for(i in state){print i,state[i]}}' ##统计连接

**四、apache原理讲解：**

1.概述：apache作为最早的web服务程序，基于http协议提供网页浏览服务；

2.特点：模块化设置、开放源代码、跨平台应用、支持多种web编程语言、运行稳定；

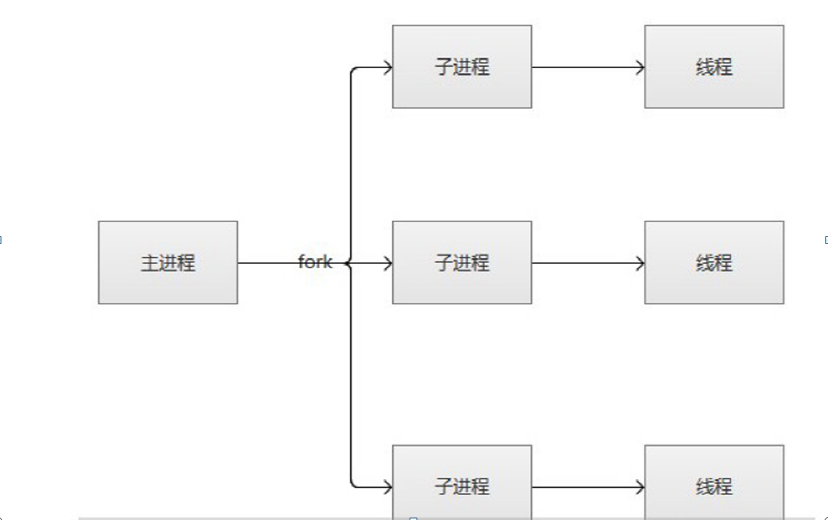
3.apache的常见三种**工作模式**：

**Ap**ache 的核心模块叫多路处理模块Multi-Processing Module，简称 MPM；

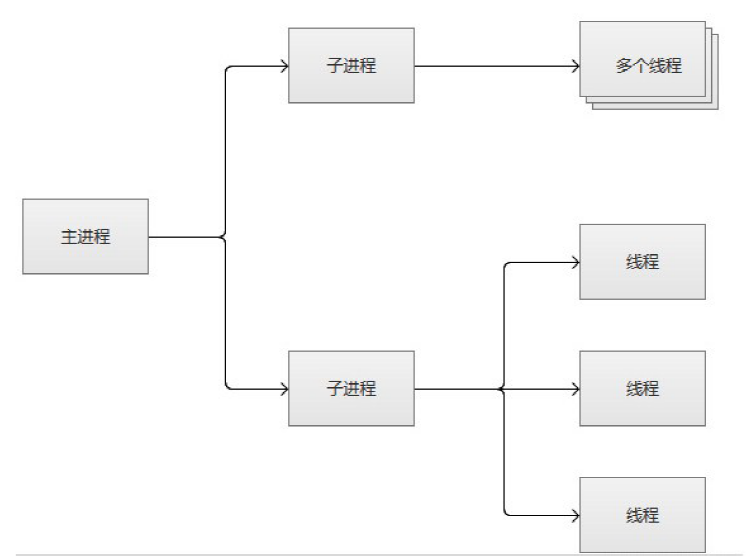
**MPM-prefork**：多进程模式，一个进程处理一个连接，每个进程相对来讲都是独立的，这个过程会用到select机制来通知；特点：稳定、响应快、消耗大量cpu和内存、keep-alive长连接占据问题；

注解：keep-alive长连接-- TCP连接在发送后将仍然保持打开状态，于是，浏览器可以继续通过相同的连接发送请求。保持连接节省了为每个请求建立新连接所需的时间，还节约了带宽。

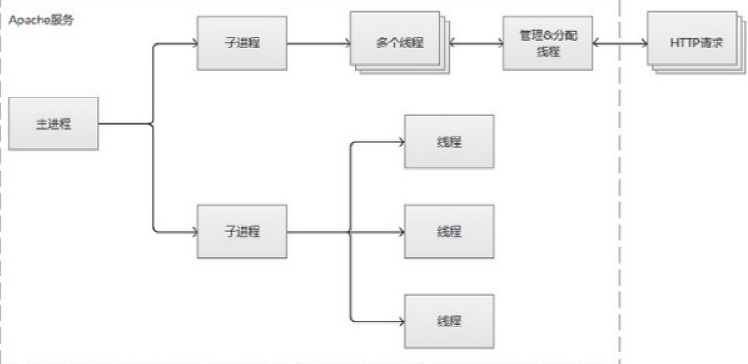
实现长连接要客户端和服务端都支持长连接。



**MPM-worker：**多进程多线程、一个进程开多个线程、每一个线程处理一个连接，但通知机制还是select不过可以接受更多的请求；特点：节省资源、兼容性不好、稳定性不高、keep-alive长连接占据问题；



**MPM-event**：worker的升级版、把服务器进程和连接进行分离，基于异步I/O模型。请求过来后进程并不处理请求，而是直接交由其他机制来处理，通过epoll机制来通知请求是否完成；在这个过程中，进程本身一直处于空闲状态，可以一直接收用户请求。可以实现一个进程响应多个用户请求。并且event模式对于keep-alive连接处理也有所优化，event模式有单独的线程处理keep-alive长连接，执行完毕后，又允许它释放。这样增强了高并发场景下的请求处理能力。特点：支持海量级高并发负载、消耗资源少、但其对https支持的不完美；



**总结：**

不同模式效率不同，主要用于调优、命令“httpd -V”查看当前使用模式；

httpd\_2.2版本默认的模式为prefork，httpd\_2.4版本默认的模式为event；

**五、案例：搭建apache网站服务；**

**环境：**

|  |  |  |  |
| --- | --- | --- | --- |
| 系统 | IP地址 | 主机名 | 所需软件 |
| Centos 6.5 | 192.168.100.150 | www.linuxfan.cn | httpd-2.2.17.tar.gz |

**实验步骤：**

* 准备环境，上传软件包，并解压编译安装；
* 安装后优化调整；
* 修改定制配置文件并启动服务；
* 客户端访问测试；
* 优化httpd服务：调整其工作模式；
* **准备环境，上传软件包，并解压编译安装；**

[root@www ~]# rpm -e httpd --nodeps

[root@www ~]# wget http://archive.apache.org/dist/httpd/httpd-2.2.17.tar.gz

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

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

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

[root@www httpd-2.2.17]# echo $? ##返回0为成功

0

[root@www httpd-2.2.17]# ls /usr/local/httpd/ ##安装成功

bin cgi-bin error icons lib man modules

build conf htdocs include logs manual

注解：

bin：存放命令；

lib：httpd的链接文件

man：帮助文档

modules：模块文件

conf：配置文件存放位置

htdocs：网页存放位置

logs：日志存放文件位置

[root@www httpd-2.2.17]# cd

[root@www ~]#

* **安装后优化调整；**

[root@www ~]# ln -s /usr/local/httpd/bin/\* /usr/local/bin/ ##优化执行命令的路径

[root@www ~]# cp /usr/local/httpd/bin/apachectl /etc/init.d/httpd

[root@www ~]# vi /etc/init.d/httpd

1 #!/bin/bash

2 # chkconfig: 35 85 15 ##在3和5运行级别开机启动，开机启动顺序为85，关机关闭顺序为15

81 $HTTPD -k $ARGV &&echo "httpd is $ARGV complete." ##第82行，添加输出信息

:wq

[root@www ~]# chmod +x /etc/init.d/httpd

[root@www ~]# chkconfig --add httpd

[root@www ~]# chkconfig --list httpd

httpd 0:关闭 1:关闭 2:关闭 3:启用 4:关闭 5:启用 6:关闭

* **修改定制配置文件、准备网页文档、并启动服务；**

[root@www ~]# vi /usr/local/httpd/conf/httpd.conf ##修改主配置文件

:set nu ##打印出行号

29 ServerRoot "/usr/local/httpd"

40 Listen 80

65 User daemon

66 Group daemon

88 ServerAdmin admin@linuxfan.cn ##修改管理员的邮箱

97 ServerName www.linuxfan.cn:80 ##添加网站的域名

104 DocumentRoot "/usr/local/httpd/htdocs"

186 ErrorLog "logs/error\_log"

193 LogLevel warn

215 CustomLog "logs/access\_log" common

:wq

[root@www ~]# cat <<END > /usr/local/httpd/htdocs/index.html ##准备网页文档

<html>

<head>

<title>www.linuxfan.cn</title>

</head>

<body>

<h1><span style="color:#3F9; font-size:36px;">www.linuxfan.cn</h1>

<p><img src="./linux.jpg" alt="www.linuxfan.cn" title="www.linuxfan.cn width=700"" height="500" /></p>

</body>

</html>

END

[root@lwh ~]# ls /usr/local/httpd/htdocs/ ##上传网页中调用的图片

index.html linux.jpg

[root@www ~]# /etc/init.d/httpd start

httpd is start complete.

[root@www ~]# netstat -utpln |grep httpd

tcp 0 0 :::80 :::\* LISTEN 73341/httpd

[root@www httpd]# netstat -nat | awk '/^tcp/ {++S[$NF]} END {for(key in S) print key,"\t",S[key]}'

FIN\_WAIT2 38

CLOSING 3

SYN\_RECV 1

CLOSE\_WAIT 1

TIME\_WAIT 261

ESTABLISHED 71

LAST\_ACK 2

FIN\_WAIT1 9

注解：

CLOSED 无连接是活动的或正在进行

LISTEN 服务器在等待进入呼叫

SYN\_RECV 一个连接请求已经到达，等待确认

SYN\_SENT 应用已经开始，打开一个连接

ESTABLISHED 正常数据传输状态/当前并发连接数

FIN\_WAIT1 应用说它已经完成

FIN\_WAIT2 另一边已同意释放

ITMED\_WAIT 等待所有分组死掉

CLOSING 两边同时尝试关闭

TIME\_WAIT 另一边已初始化一个释放

LAST\_ACK 等待所有分组死掉

* **客户端访问测试；**

[root@www ~]# yum -y install elinks

[root@www ~]# elinks --dump http://192.168.100.150

www.linuxfan.cn

[1]www.linuxfan.cn

References

Visible links



* **优化httpd服务：调整其工作模式；**

**1.查看工作模式：**

[root@www ~]# /usr/local/httpd/bin/httpd -V |grep MPM ##查看工作模式

Server MPM: prefork

**注：设置httpd的工作模式时，不仅考虑网站的并发连接等工作效率，也要考虑服务器硬件占用情况，内存尤为重点考虑；**

**2.prefork（进程+子进程）模式，默认：**

|  |  |
| --- | --- |
| StartServers | 服务器启动时建立的子进程数量，推荐：小型网站：默认、中：20-50、大：50-100； |
| MinSpareServers | 空闲子进程最小数量，当空闲子进程数少于该数值将产生新的子进程，推荐：与startservers一致； |
| MaxSpareServers | 空闲子进程最大数量，当空闲子进程数大于该数值将杀死多于子进程。这个值不要设得过大，但如果设的值比 MinSpareServers 小， Apache 会自动把其调整为MinSpareServers+1； |
| MaxClients | 限定最大用户并发数， 是对 Apache 性能影响最大的参数。默认是150个进程，如果请求总数达到该设置的值时--可以使用ps -ef |grep http |wc -l来查看，那么后边的请求就需要排队，直到某个已经建立完成的请求处理完成后。这就是apache服务器系统资源剩余很多，但是http访问还是很慢的原因，理论上这个值设置的越大，apache可以处理的请求就越多，那么如果该值超过256，那么apache则无法启动，如果想配置更大值，必须在源码包中修改serverlimit（服务极限）这个参数，该参数最大为200000，对位于源代码包下 server/mpm/prefork/prefork.c 中以下两行做相应修改即可：  #define DEFAULT\_SERVER\_LIMIT 256  #define MAX\_SERVER\_LIMIT 200000  改后重新编译安装； |
| MaxRequestsPerChild | 每个子进程在其生存期内允许服务的最大请求数量，若为0则代表该进程不会因为请求数量的变化而停止，建议10000-30000，避免出现活动请求占用服务器内存资源； |

[root@www ~]# /usr/local/httpd/bin/httpd -V |grep MPM

Server MPM: prefork

-D APACHE\_MPM\_DIR="server/mpm/prefork"

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

36 <IfModule mpm\_prefork\_module>

37 StartServers 5

38 MinSpareServers 5

39 MaxSpareServers 20

40 MaxClients 150

41 MaxRequestsPerChild 10000

42 </IfModule>

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

370 Include conf/extra/httpd-mpm.conf

[root@www ~]# /etc/init.d/httpd restart

[root@www ~]# ps aux |grep httpd |grep -v grep ##一个主进程，五个子进程

daemon 3849 0.0 0.3 27996 1528 ? S 21:37 0:00 /usr/local/httpd/bin/httpd -k restart

daemon 3850 0.0 0.3 27996 1528 ? S 21:37 0:00 /usr/local/httpd/bin/httpd -k restart

daemon 3851 0.0 0.3 27996 1528 ? S 21:37 0:00 /usr/local/httpd/bin/httpd -k restart

daemon 3852 0.0 0.3 27996 1528 ? S 21:37 0:00 /usr/local/httpd/bin/httpd -k restart

daemon 3853 0.0 0.3 27996 1528 ? S 21:37 0:00 /usr/local/httpd/bin/httpd -k restart

root 53394 0.0 0.4 27996 2088 ? Ss 19:19 0:00 /usr/local/httpd/bin/httpd -k restart

**3.worker（多进程+多线程）工作模式：**

|  |  |
| --- | --- |
| StartServers | 服务器启动时建立的子进程数量，推荐：小=默认、中=3-5、大=5-10； |
| MaxClients | 限定最大用户并发数、推荐：500以上**；** |
| MinSpareThreads | 空闲子线程最小数量，当空闲子线程数少于该数值将产生新的子线程，推荐：小=默认、中=50-100、大=100-200； |
| MaxSpareServers | 空闲子线程最大数量，当空闲子线程数大于该数值将杀死多于子线程，推荐：小=默认、中=80-160、大=200-400； |
| ThreadsPerChild | 每个子进程的最大并发线程数，推荐：小型网站1000、中1000-2000、大2000-3500，推荐：小=默认、中=50-100、大=100-200； |
| MaxRequestsPerChild | 每个子进程在其生存期内允许服务的最大请求数量。如果设置为“ 0”，子进程将永远不会结束。推荐：小10000、大20000-50000； |
| 注：如若开启worker工作模式，需要在编译安装时--with-mpm=worker选项指定；  当从prefork切换到worker时，重新编译安装时；  执行make命令出现以下错误：  /usr/src/httpd-2.2.17/server/mpm/prefork/prefork.c:862: undefined reference to `ap\_mpm\_pod\_signal'  collect2: ld returned 1 exit status  make[1]: \*\*\* [httpd] 错误 1  make[1]: Leaving directory `/usr/src/httpd-2.2.17'  make: \*\*\* [all-recursive] 错误 1  可执行命令make clean清空已编译内容，重新编译即可； | |

[root@www ~]# /usr/local/httpd/bin/httpd -V |grep MPM

Server MPM: worker

-D APACHE\_MPM\_DIR="server/mpm/worker"

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

51 <IfModule mpm\_worker\_module>

52 StartServers 2

53 MaxClients 150

54 MinSpareThreads 25

55 MaxSpareThreads 75

56 ThreadsPerChild 50

57 MaxRequestsPerChild 10000

58 </IfModule>

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

371 Include conf/extra/httpd-mpm.conf

:wq

[root@www ~]# ps aux |grep httpd |grep -v grep

root 33118 0.0 0.4 28168 2044 ? Ss 00:18 0:00 /usr/local/httpd/bin/httpd -k restart

daemon 33119 0.0 0.2 27900 1372 ? S 00:18 0:00 /usr/local/httpd/bin/httpd -k restart

daemon 33121 0.0 0.4 628528 2156 ? Sl 00:18 0:00 /usr/local/httpd/bin/httpd -k restart

**4.event工作模式：**

|  |  |
| --- | --- |
| ServerLimit | 服务器开启最大的进程数 |
| StartServers | 服务器启动时建立的子进程数量，推荐：小=默认、中=3-5、大=5-10； |
| MaxClients | 限定最大用户并发数、为ServerLimit\* ThreadsPerChild、推荐：500以上**；** |
| MinSpareThreads | 空闲子线程最小数量，当空闲子线程数少于该数值将产生新的子线程，推荐：小=默认、中=50-100、大=100-200； |
| MaxSpareServers | 空闲子线程最大数量，当空闲子线程数大于该数值将杀死多于子线程，推荐：小=默认、中=80-160、大=200-400； |
| ThreadsPerChild | 每个子进程的最大并发线程数，推荐：小型网站1000、中1000-2000、大2000-3500，推荐：小=默认、中=50-100、大=100-200； |
| MaxConnectionsPerChild | 每个子进程在其生存期内允许服务的最大请求数量。如果设置为“ 0”，子进程将永远不会结束。推荐：小10000、大20000-50000； |
| 注：如若开启event工作模式，需要在编译安装时--with-mpm=event选项指定，并且内核版本最低需要2.6； | |

[root@www ~]# /usr/local/httpd/bin/httpd -V |grep MPM

Server MPM: Event

-D APACHE\_MPM\_DIR="server/mpm/experimental/event"

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

111 <IfModule mpm\_event\_module>

112 ServerLimit 1000

113 StartServers 20

114 MinSpareThreads 25

115 MaxSpareThreads 1200

116 ThreadsPerChild 50

117 MaxConnectionsPerChild 1000

118 </IfModule>

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

371 Include conf/extra/httpd-mpm.conf

:wq

[root@www ~]# ps aux |grep httpd |grep -v grep

root 18598 0.1 3.6 44416 18276 ? Ss 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18599 0.0 0.2 27900 1376 ? S 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18600 0.0 0.4 644776 2164 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18601 0.0 0.4 644776 2156 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18602 0.0 0.4 644776 2156 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18603 0.0 0.4 644776 2152 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18604 0.0 0.4 644776 2160 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18605 0.0 0.4 644776 2152 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18606 0.0 0.4 644776 2156 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18607 0.0 0.4 644776 2156 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18608 0.0 0.4 644776 2160 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18609 0.0 0.4 644776 2152 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18610 0.0 0.4 644776 2152 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18611 0.0 0.4 644776 2152 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18612 0.0 0.4 644776 2156 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18613 0.0 0.4 644776 2156 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18614 0.0 0.4 644776 2152 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18615 0.0 0.4 644776 2156 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18616 0.0 0.4 644776 2156 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18617 0.0 0.4 644776 2160 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18618 0.0 0.4 644776 2156 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start

daemon 18619 0.0 0.4 644776 2152 ? Sl 00:36 0:00 /usr/local/httpd/bin/httpd -k start