apache配置详解(很详细)

二、 httpd.conf中常用参数介绍

1． 连接类参数

l TimeOut

服务器在断定请求失败前等待的时间,即接收和发送数据的超时时间。

【说明】

TimeOut参数用于设置Apache在等待以下三种事件的时间：

u 接受一个GET请求耗费的总时间。

u POST或PUT请求时，接受两个TCP包之间的时间。

u 回应时TCP包传输中两个ACK包之间的时间。

还可以理解成Apache允许每次通过HTTP协议传输数据的最大时间。

l KeppAlive

启用HTTP持续作用。

【说明】

持续作用扩展自HTTP/1.0和HTTP/1.1的长连接特性。提供了长效的HTTP会话，用以在同一个TCP连接中进行多次请求。在某些情况下，这样的方式会对包含大量图片的HTML文件起到50%的加速作用。对于使用HTTP/1.0的客户端来说，只有客户端设置使用的时候才会使用持续作用连接。而对于HTTP/1.1的客户端来说，如果没有进行特殊指定，长连接将是默认的连接方式。

l MaxKeppAliveRequests

一个长连接中允许的请求数量。

【说明】

该参数限制了当启用KeepAlive时，每次连接允许的请求数量。如果将此值设为0，将不限制请求的数量。这里建议最好将此值设为一个比较大的值，以确保最优的服务器性能。

l KeppAliveTimeOut

持续作用中服务器在两次请求之间的等待时间。

【说明】

Apache在关闭本次连接前等待下一次请求的时间，即在这段时间内该连接没有接收到请求就会关闭此连接。一旦收到一个请求，超时值将会被设置为KeppAliveTimeOut的值。

注意：对于高负荷的服务器来说，如果把该参数的值设置的较大可能会导致一些性能方面的问题，因为KeppAliveTimeOut的值会影响释放空闲进程、线程时间的数量，如果该值大，那么在一定时间区间内释放出来的空闲进程、线程

的数量会少于该值小的，所以会降低服务器处理请求的能力，从而影响整个系统的吞吐量。

l Listen

服务器监听IP地址和端口。

【说明】

Listen参数是指Apache服务器在指定的IP地址和端口上进行监听；默认情况下Apache会在所有IP地址上监听。Listen是一个必须设置的指令。如果在配置文件中找不到这个指令，服务器将无法启动。

Listen参数还可以指定服务器在哪个端口或地址和端口的组合上进行监听请求。如果只指定一个端口，服务器将在所有地址上监听该端口。如果指定了地址和端口的组合，服务器将按照指定地址和指定的端口进行监听。

使用多个Listen参数可以指定多个不同的监听端口和/或地址端口组合。

例如，想要服务器接受80和8080端口上的请求，可以这样设置：

Listen 80

Listen 8080

为了让服务器在两个确定的地址端口组合上接受请求，可以这样设置：

Listen 192.64.182.1:80

Listen 192.64.182.5:8080

注意：多个Listen指令指定了同一个地址和端口的组合后，会导致"Address already in use"错误。

2． 系统路径管理类参数

l ServerRoot

服务器的安装基础目录。

【说明】

该参数设置了服务器所在的目录。一般来说它将包含conf/和logs/子目录。其它配置文件的相对路径都基于此目录 (比如Include或LoadModule)。

例如：

ServerRoot /etc/httpd

l DocumentRoot

组成网络上可见的主文档树的根目录。

【说明】

此参数设置了httpd服务的目录。在没有配置类似Alias这种参数的情况下，服务器会将请求中的URL附加到DocumentRoot后面以构成指向文档的路径。比如说：

DocumentRoot /etc/httpd/www/web

于是对http://www.ccb.com.cn/index.html的访问就会指向/etc/httpd/www/web/index.html。如果参数中不是绝对路径，则被假定为是相对于ServerRoot的路径。

注意：指定DocumentRoot时不应包括最后的"/"。

l Directory

可以封装一组参数，使之仅对文件空间中的某个目录及其子目录生效

【语法】

<Directory directory-path> ... </Directory>

【说明】

<Directory>和</Directory>用于封装一组参数，使其对某个目录及其子目录生效。directory-path可以是一个目录的完整路径，或是包含了Unix shell匹配语法的通配符字符串，但是通配符都不能匹配"/"字符，例如：<Directory /\*/public\_html>是无法匹配/home/user/public\_html 的，而<Directory /home/\*/public\_html>却能够正确匹配。

directory-path参数必须与被访问文件所在文件系统的路径保持一致。如果有多个非正则表达式，<Directory>配置段符合并包含某文档的目录(或其父目录)，那么会以短目录优先的规则进行应用。<Directory />的默认访问权限为"Allow from All"，这意味着A

pache没有进行访问控制，通过设置Order，Deny，Allow，AllowOverride这个几个参数可以对访问进行控制。

下面简单介绍一下这4个参数的用法。

1> Allow

该参数是控制哪些主机才可以访问目标。

示例：

Allow from 192.64.182.53

Allow from 192.64.182.1 192.64.182.3

表示IP地址为192.64.182.1或192.64.182.3或192.64.182.53才可以访问目标。

2> Deny

该参数是控制哪些主机被禁止访问目标。

示例：

Deny from 192.64.182.53

Deny from 192.64.182.1 192.64.182.3

表示IP地址为192.64.182.1或192.64.182.3或192.64.182.53则不能访问目标。

3>Order

Order参数是控制Allow和Deny参数生效顺序的，常用的取值是：Deny,Allow 和Allow,Deny。例如：

Order Deny,Allow

Deny from 192.64.182.53

Allow from 192.64.182.1 192.64.182.3

表示先考虑Deny条件再考虑Allow条件，该配置的意思是拒绝IP地址为192.64.182.53的访问，只允许192.64.182.1 和192.64.182.3的访问。

再看一个例子：

Order Allow,Deny

Allow from all

Deny from 192.64.182.53

表示只拒绝IP地址为192.64.182.53的访问。

4>AllowOverride

当服务器发现一个.htaccess文件(由AccessFileName指定)时，它需要知道在这个文件中声明的哪些指令能覆盖在此之前指定的配置参数。一般情况下NONE即可。

【Directory参数小结】

最后给出一个完整封装目录的配置段：

<Directory "/home/hrdc/images">

Options Indexes FollowSymLinks //对URL映射到的系统目录产生文件列表

AllowOverride None

Order Deny,Allow

Allow from all

</Directory>

上面的配置对系统中的"/home/hrdc/images"目录进行了封装，而且对访问不加任何限制。这段配置后面在讲如何将静态文件放置到Apache上还会用到。

3．监控反馈类参数

l PidFile

服务器用于记录父进程(监控进程)PID的文件

【说明】

PidFile指令设置服务器用于记录

父进程(监控进程)PID的文件。如果指定的不是绝对路径，那么将视为基于ServerRoot的相对路径。

示例：

PidFile /var/run/apache.pid

这个文件通常用来给服务器父进程发送一个信号，用于关闭或重启服务器，以便重新打开ErrorLog和TransferLog文件、重新读取配置文件。

l ServerAdmin

服务器返回给客户端的错误信息中所包含的管理员邮件地址。

【说明】

该参数是在所有返回给客户端的错误信息中给出管理员的邮件地址。但也可以是一个URL地址，如果httpd不能将该参数的值识别为URL，它就会假定它是一个email-address ，并在超连接中用在mailto后面。这里推荐配置一个Email地址,如果配置的是URL一定要保证指向一个受控制的服务器，否则用户将无法确保和管理员取得联系。

示例：

ServerAdmin hradmin.zh@ccb.cn

4．日志管

理类参数：

l LogLevel

控制错误日志的级别

【说明】

LogLevel用于设置服务器按照日志级别来记录日志信息。该参数可以选择的level有：

Level

描述

例子

emerg

紧急(系统无法使用)

"Child cannot open lock file. Exiting"

alert

必须立即采取措施

"getpwuid: couldn't determine user name from uid"

crit

致命情况

"socket: Failed to get a socket, exiting child"

error

错误情况

"Premature end of script headers"

warn

警告情况

"child process 1234 did not exit, sending another SIGHUP"

notice

一般重要情况

"httpd: caught SIGBUS, attempting to dump core in ..."

info

普通信息

"Server seems busy, (you may need to increase StartServers, or Min/MaxSpareServers)..."

debug

调试信息

"Opening config file ..."

注意：当指定了某个级别后，所有级别高于它的信息也会被同时记录。建议至少使用crit级别。当错误日志是一个单独分开的正式文件的时候，notice级别的消息总是会被记录下来，而不能被屏蔽。

l ErrorLog

存放错误日志的位置

【说明】

该参数指定了当服务器遇到错误时记录日志的文件名。如果该值不是一个以斜杠(/)开头的绝对路径，那么将被认为是一个相对于ServerRoot的相对路径。

示例

ErrorLog /etc/var/log/httpd/error\_log

如果配置了一个以管道符号(|)开头的值，那么会为它指定一个命令来处理错误日志。

示例

ErrorLog "|/usr/local/bin/httpd\_errors"

注意：当在非Unix平台上输入文件路径的时候，路径分隔符必须统一使用正斜线(/)。

l CustomLog

设置服务器访问日志的文件名和格式。

【说明】

该参数用来对服务器的请求进行日志记录。第一个参数指定了日志文件的位置，第二个参数用于设置日志的格式。

示例：

CustomLog logs/access\_log "%h %l %u %t ""%r"" %>s %b"

l 定制日志文件格式

LogFormat和CustomLog的格式化参数是一个字符串。这个字符串会在每次请求发生的时候，被记录到日志中去。它可以包含将被原样写入日志文本放入字符串以及C风格的控制字符""n"和""t"。文本中的引号和反斜杠应通过"""来转义。请求本身的情况也将通过在格式字符串中放置各种"%"转义符的方法来记录，它们在写入日志文件时，根据下表的定义进行转换：

格式字符串

描述

%%

百分号(Apache2.0.44或更高的版本)

%a

远端IP地址

%A

本机IP地址

%B

除HTTP头以外传送的字节数

%b

以CLF格式显示的除HTTP头以外传送的字节数，也就是当没有字节传送时显示'-'而不是0。

%{Foobar}C

在请求中传送给服务端的cookieFoobar的内容。

%D

服务器处理本请求所用时间，以微为单位。

%{FOOBAR}e

环境变量FOOBAR

的值

%f

文件名

%h

远端主机

%H

请求使用的协议

%{Foobar}i

发送到服务器的请求头Foobar:的内容。

%l

远端登录名(由identd而来，如果支持的话)，除非IdentityCheck设为"On"，否则将得到一个"-"。

%m

请求的方法

%{Foobar}n

来自另一个模块的注解Foobar的内容。

%{Foobar}o

应答头Foobar:的内容。

%p

服务器服务于该请求的标准端口。

%P

为本请求提供服务的子进程的PID。

%{format}P

服务于该请求的PID或TID(线程ID)，format的取值范围为：pid和tid(2.0.46及以后版本)以及hextid(需要APR1.2.0及以上版本)

%q

查询字符串(若存在则由一个"?"引导，否则返回空串)

%r

请求的第一行

%s

状态。对于内部重定向的请求，这个状态指的是原始请求的状态，---%>s则指的是最后请求的状态。

%t

时间，用普通日志时间格式(标准英语格式)

%{format}t

时间，用strftime(3)指定的格式表示的时间。(默认情况下按本地化格式)

%T

处理完请求所花时间，以秒为单位。

%u

远程用户名(根据验证信息而来；如果返回status(%s)为401，可能是假的)

%U

请求的URL路径，不包含查询字符串。

%v

对该请求提供服务的标准ServerName。

%V

根据UseCanonicalName指令设定的服务器名称。

%X

请求完成时的连接状态：

X=

连接在应答完成前中断。

+=

应答传送完后继续保持连接。

-=

应答传送完后关闭连接。

(在1.3以后的版本中，这个指令是%c，但这样就和过去的SSL语法：%{var}c冲突了)

%I

接收的字节数，包括请求头的数据，并且不能为零。要使用这个指令你必须启用mod\_logio模块。

%O

发送的字节数，包括请求头的数据，并且不能为零。要使用这个配置你必须启用mod\_logio模块。

修饰符

可以紧跟在"%"后面加上一个逗号分隔的状态码列表来限制记录的条目。例如，"%400,501{User-agent}i"只记录状态码400和501发生时的User-agent头内容；不满足条件时用"-"代替。状态码前还可以加上"!"前缀表示否定，"%!200,304,302{Referer}i"记录所有不同于200,304,302的状态码发生时的Referer头内容。"<"和">"修饰符可以用来指定对于已被内部重定向的请求是选择原始的请求还是选择最终的请求。默认情况下，%s, %U, %T, %D, %r 使用原始请求，而所有其他格式串则选择最终请求。例如，%>s 可以用于记录请求的最终状态，而 %<u 则记录一个已经被内部重定向到非认证资源的请求的原始认证用户。

官方的一些说明

出于安全考虑，从2.0.46版本开始，%r, %i, %o 中的特殊字符，除了双引号(")和反斜线(")分别用 "" 和 "" 进行转义、空白字符用C风格("n, "t 等)进行转义以外，非打

印字符和其它特殊字符使用 "xhh 格式进行转义(hh是该字符的16进制编码)。在2.0.46以前的版本中，这些内容会被完整的按原样记录。这种做法将导致客户端可以在日志中插入控制字符，所以在处理这些日志文件的时候要特别小心。在2.0版本中(不同于1.3)，%b 和 %B 格式字符串并不表示发送到客户端的字节数，而只是简单的表示HTTP应答字节数(在连接中断或使用SSL时与前者有所不同)。mod\_logio提供的 %O 格式字符串将会记录发送的实际字节数。

示例

一些常见的格式串：

通用日志格式(CLF)

"%h %l %u %t ""%r"" %>s %b"

带虚拟主机的通用日志格式

"%v %h %l %u %t ""%r"" %>s %b"

NCSA扩展/组合日志格式

"%h %l %u %t ""%r"" %>s %b ""%{Referer}i"" ""%{User-agent}i"""

Referer日志格式

"%{Referer}i -> %U"

Agent(Browser)日志格式

"%{User-agent}i"

URL映射类参数

l Alias

将URL映射到文件系统的特定区域。

【说明】

语法：Alias URL-path file-path|directory-path

Alias参数使文件可以被存储在DocumentRoot以外的本地文件系统中。以(%已解码的)url-path路径开头的URL可以被映射到以directory-path开头的本地文件中。

示例：

Alias /image /etc/var/www/images

将不会被匹配。

对"http://www.ccb.com/image/foo.gif"的请求，服务器将返回"/etc/var/www/images/foo.gif"文件。由于该参数是匹配完整路径，所以请求是"http:// www.ccb.com /imagefoo.gif"

注意：如果url-path中有后缀"/"，则服务器要求有后缀"/"以扩展此别名。也就是说"Alias /icons/ /usr/local/apache/icons/"并不能对"/icons"实现别名.

注意，可能需要额外指定一个<Directory>段来覆盖别名的最终对象。由于只有出现在<Directory>段之前的别名才会被检测，所以它只对最终对象生效。如果对在DocumentRoot之外的某个目录建立了一个Alias ，则可能需要明确的对目标目录设定访问权限。

示例：

Alias /image /ftp/pub/image

<Directory /ftp/pub/image>

Order allow,deny

Allow from all

</Directory>

二、多路处理模块的配置说明--prefork.c模块(一个非线程型的、预派生的MPM)、worker.c模块(支持混合的多线程多进程的多路处理模块)

Apache HTTP服务器是一个强大的、灵活的能够在多种平台、不同环境下运行的Web服务器。由于不同的平台和不同的环境经常产生不同的需求，为了达到同样的最佳效果则需要采取不同的实现方法， Apache的模块化设计就可以很好的适应大量不同的环境。使得网站管理员能够在编译和运行时凭借载入不同的模块来决定服务器的附加功能。Apache的多路处理模块(MPM)就是用于选择处理网络端口绑定、接受请求并指派子进程处理来自客户端的请求。

默认的MPM

下表列出了不同操作系

统上默认的MPM。如果编译时没有进行选择，这将是默认选择的MPM。

操作系统名称

MPM名称

BeOS

beos

Netware

mpm\_netware

OS/2

mpmt\_os2

Unix

prefork

Windows

mpm\_winnt

1、prefork.c模块(一个非线程型的、预派生的MPM)

prefork.c模块是由一个单独的控制进程(父进程)负责产生子进程，这些子进程用于监听请求并作出应答。Apache设置了一些备用的(spare)或者是空闲的子进程来处理即将接收的请求，这样可以避免服务器接收到请求后在创建子进程。在Unix系统中，父进程通常以root身份运行以便邦定80端口，而 Apache产生的子进程通常以一个低特权的用户运行。User和Group参数就是用于设置子进程的低特权用户。运行子进程的用户必须要对它所服务的内容有读取的权限，但是对服务内容之外的其他资源最好拥有尽可能少的权限。

【配置示例】

<IfModule prefork.c>

StartServers 8

MinSpareServers 5

MaxSpareServers 20

ServerLimit 400

MaxClients 256

MaxRequestsPerChild 4000

</IfModule>

【参数说明】

1.ServerLimit

默认的MaxClient最大是256个线程,如果想设置更大的值，就需要修改ServerLimit这个参数。例子中的400是ServerLimit这个参数的最大值。如果需要更大，则必须编译apache,此前都是不需要重新编译Apache。

2．StartServers

指定服务器启动时建立的子进程数量，因为子进程的数量动态的取决于负载的轻重，所以一般没有必要调整这个参数，prefork模式默认为5。

3．MinSpareServers

指定空闲子进程的最小数量，所谓空闲子进程是指没有正在处理请求的子进程。默认为5。如果当前空闲子进程数少于MinSpareServers ，那么Apache将以最大每秒一个的速度产生新的子进程，只有机器在非常繁忙的情况下才需要调整这个参数。

4．MaxSpareServers

设置空闲子进程的最大数量。默认为10。如果当前有超过MaxSpareServers数量的空闲子进程，那么父进程将杀死多余的子进程。如果该参数的值设置比MinSpareServers小，Apache则会自动将其修改成"MinSpareServers+1"。

5．MaxClients

指可以服务于客户端请求的最大子进程数量，即限定同一时间客户端最大接入请求的数量，默认值为256。任何超过MaxClients限制的请求都将进入等候队列,一旦一个连接被释放，队列中的请求将得到服务。

6．MaxRequestsPerChild

每个子进程在其生存期内允许处理的最大请求数，默认为10000.到达MaxRequestsPerChild的限制后，子进程将会结束。如果MaxRequestsPerChild为"0"，子进程将永远不会结束。这个参数也可以理解成控制服务器杀死旧进程产生新进程的频率。

注意：

从系统稳定性来考虑将MaxRequestsPer

Child设置成非零有两个好处：

1.可以防止(偶然的)内存泄漏无限进行，从而耗尽内存。

2.给进程一个有限寿命，从而有助于当服务器负载减轻的时候减少活动进程的数量。

【工作原理介绍】

首先服务启动后会创建以StartServers个数的进程，然后等待来自客户端的请求。我们这里先假设从客户端来了大量的请求，这时Apache服务器会根据自身的负载情况自动创建新进程，如果服务器一直没有空闲进程那么它就会一直创建新进程，直到满足MaxClients和ServerLimit设置的最大值。如果来自客户端的负载没有那么大，Apache服务器将会根据MinSpareServers、MaxSpareServers和MaxRequestsPerChild设置的值来杀掉多余的进程。其中每个进程在某个确定的时间只能维持一个连接。

【小结】

上面叙述的这些参数中，对系统性能影响较大的有两个：MaxClients 和ServerLimit。这个两个参数主要影响Web服务器处理客户端请求的能力，它们决定着服务器创建用来处理请求的进程的最大值。如果来自客户端的请求总数已经达到系统创建进程的最大值（可通过ps -ef|grep http|wc –l来确认），那么后面来的请求就要排队，直到某个已处理请求完成。这就是应用系统资源还很富余而HTTP访问却很慢的主要原因。如何找出这个两个参数的最佳值需要综合很多因素，但一般情况下可以参考系统性能测试结果和Web服务器的系统资源。

注意：prefork模式下创建较多的进程将会占去大量系统内存，如果MaxClients和ServerLimit设置过大时可能会造成Web服务器崩溃。

2、worker.c模块(支持混合的多线程多进程的多路处理模块)

worker 模块使用多个子进程，每个子进程有多个线程。每个线程在某个确定的时间只能维持一个连接。通常来说，在一个高流量的HTTP服务器上，worker 模式是个比较好的选择，因为它的内存使用比prefork要低得多。但worker模式也有不完善的地方，如果一个线程崩溃，整个进程就会连同其所有线程一起"死掉"。由于线程共享内存空间，所以一个进程在运行时必须被系统识别为"每个线程都是安全的"。

【配置示例】

<IfModule worker.c>

StartServers 2

MaxClients 400

ServerLimit 12

MinSpareThreads 25

MaxSpareThreads 75

ThreadsPerChild 25

ThreadLimit 75

MaxRequestsPerChild 0

</IfModule>

【参数说明】

1.ServerLimit

服务器允许配置的进程数上限。这个参数和ThreadLimit结合使用便决定了MaxClients所能设置的最大值。任何在重启期间对这个参数的改变都将被忽略，但对MaxClients的修改却会生效。

2.ThreadLimit

每个子进程可设置的线程数上限，这个参数决定了每个子进程可创

建线程的数，即ThreadsPerChild的上限。任何在重启期间对这个参数的改变都将被忽略，但对ThreadsPerChild的修改却会生效。默认值是"64".

3.StartServers

服务器启动时建立的子进程数，默认值是"3"。

4.MinSpareThreads

最小空闲线程数,默认值是75。MPM将基于整个服务器监视空闲线程数。如果服务器中总的空闲线程数太少，子进程将产生新的空闲线程。

5.MaxSpareThreads

设置最大空闲线程数。默认值是250。MPM将基于整个服务器监视空闲线程数。如果服务器中总的空闲线程数太多，子进程将杀死多余的空闲线程。MaxSpareThreads的取值范围是有限制的，在Apache的 worker模式下是要求大于等于 MinSpareThreads与ThreadsPerChild之和来自动修正你设置的值。

6.MaxClients

允许同时接收客户端最大请求的数量(最大线程数量)。任何超过MaxClients限制的请求都将进入等候队列。默认值是400,即16 (ServerLimit)乘以25(ThreadsPerChild)。因此需要增加MaxClients的时候，你必须同时增加 ServerLimit的值。

7.ThreadsPerChild

每个子进程建立常驻的执行线程数。默认值是25。子进程在启动时建立这些线程后就不再建立新的线程了。

8.MaxRequestsPerChild

设置每个子进程在其生存期内允许提供服务的最大请求数量。到达MaxRequestsPerChild的限制后，子进程将会结束。如果MaxRequestsPerChild为"0"，子进程将永远不会结束。

将MaxRequestsPerChild设置成非零值有两个好处：

1.可以防止(偶然的)内存泄漏无限进行，从而耗尽内存。

2.给进程一个有限寿命，从而有助于当服务器负载减轻的时候减少活动进程的数量。

注意

对于KeepAlive链接，只有第一个请求会被计数。事实上，它改变了每个子进程限制最大链接数量的行为。

【工作原理介绍】：

worker的工作原理是：先由主控制进程创建“StartServers”个子进程，每个子进程中含有“ThreadsPerChild”个线程，各个线程独立地处理来自客户端的请求。同Prefork一样，为了不在请求到来时在去创建线程，MinSpareThreads和MaxSpareThreads决定了最少和最多空闲线程数；随着负载逐渐增大，而现有子进程中的线程不能满足负载时，主控进程将按照“ServerLimit”和“MaxClients”的限制去创建新进程，如果“ServerLimit”达到上限而ServerLimit\* ThreadsPerChild <MaxClients时，那么主控进程将参考“ThreadLimit”的值去尝试增加某个进程的线程，前提是ThreadLimit> ThreadsPerChild。如果“ServerLimit”未达到上限而“MaxClients”达到上限，那么服务器将不采取任何行动。倘若负载逐渐减小，那么Apache服务器将根据实际情况去消减线程或进程。

【小结】

在worker模式下MinSpareThreads和MaxSpareThreads的最大缺省值分别

是75和250。这两个参数对Apache的性能影响并不大，可以按照实际情况做相应调节。 而ThreadsPerChild参数是最影响性能的一个，因为worker模式下所能同时处理的请求总数是由子进程总数与ThreadsPerChild之积来决定的。它的最大缺省值是64，如果负载较大，64也是不够的。这时要显式使用ThreadLimit指令，它的最大缺省值是20000。。注意，不要把这两个值设得太高，如果超过系统的处理能力，会使系统很不稳定，这个值最好参考性能测试的结果来设，同时子进程总数与ThreadsPerChild之积应该略大于MaxClinets。

三、配置静态文件

目前基于B/S结构的Web页面有动态和静态两种形式，其中动态页面需由服务器的解析器进行解析，通常还需连接数据库，进行数据库存取操作，最后形成HTML语言信息包反馈给浏览者；而静态页面，则无须解析，无须连接数据库，直接反馈给客户端就可以。

这里说的静态文件就是指在服务器端无需进行任何处理，就可以直接反馈给浏览器的文件，例如：HTML、JS，CSS、JPG、BMP等等。

将应用系统中的静态文件配置到Apache服务器上有几大好处，首先从客户角度来看Web页面的响应时间提高了，其次从系统资源来看应用服务器能专心处理动态文件，所以充分发挥了它的效能，再次从系统稳定性来看Apache服务器屏蔽了所有静态文件的请求，减轻了应用服务器的压力从而降低了由大访问量带来宕机的风险。

【配置示例1】

Alias /images/ "/home/hrdc/images/"

<Directory "/home/hrdc/images">

Options Indexes FollowSymLinks

AllowOverride None

Order allow,deny

Allow from all

</Directory>

#配置图片请求映射，与CSS,JS的配置类似。

<VirtualHost 192.64.108.2:8022>

<IfModule mod\_weblogic.c>

WebLogicCluster 192.64.96.18:8018,192.64.96.11:8011

MatchExpression /file/\*

MatchExpression \*.jsp

MatchExpression \*.jsf

MatchExpression \*.rtf

MatchExpression \*.xls

MatchExpression \*.doc

MatchExpression /console\*

</IfModule>

</VirtualHost>

#配置Webloig模块中的参数。

【关键参数说明】

l VirtualHost

虚拟主机。

【说明】

在这个参数内可以模仿一个Web服务配置众多参数，即在<VirtualHost>和</VirtualHost>中配置一组仅作用于特定虚拟主机的参数。

示例

<VirtualHost 192.64.182.53:8020>

ServerAdmin hrdc.zh@ccb.cn

DocumentRoot /home/hrdc

ServerName hrdc.ccb.cn

ErrorLog logs/host.foo.com-error\_log

TransferLog logs/host.foo.com-access\_log

</VirtualHost>

注意：每个虚拟主机必须对应不同的IP地址、端口或是不同的主机名。

<VirtualHost>中定义的监听地址只代表虚拟主机并不是指定Apache服务的监听地址。指定Apache监听地址的

参数则是Listen。

l IfModule

根据指定的模块是否启用为条件来决定是否进行处理。

【语法】

<IfModule [!]module-file|module-identifier> ... </IfModule>

【说明】

在该参数中配置的表达式为真的时候才进行处理。如果为假，所有其包含的参数都将被忽略。

<IfModule>段中的表达式可以为以下两种方式之一来表达：

· module

· !module

第一种情况表示，在<IfModule >和</IfModule>之间的配置参数仅当module被载入后才被执行。此模块可以是编译时静态链接的核心模块或是使用LoadModule指令动态载入的模块。第二种情况则表示，仅当module没有载入时才执行参数内的配置处理。

module可以是模块的标识符或者是编译模块时的文件名。在上面的例子中，mod\_weblogic.c就是编译模块时的文件名。

注意：<IfModule>配置段是可以嵌套的，从而可以实现简单的多模块测试。

【小结】

现在再看一下“配置示例1”中关于Weblogic的配置，其中第一行是用于指定集群中托管服务器的地址。而下面的“MatchExpression”表示当URL匹配上后面的表达式时，Apache服务器会把该请求转给Weblogic处理。从配置中可以看出如果有一个“http://192.23.108.2:8022/images/logon.jpg”这样的请求，它是不会转发到Weblogic集群上的。

本篇文章来源于：开发学院 http://edu.codepub.com 原文链接：