Nginx配置参数中文说明

[22nd 一, 13](http://wangying.sinaapp.com/archives/931) [chenai](http://wangying.sinaapp.com/archives/author/admin)

PS:最近在看<<高性能Linux服务器构建实战>>的Nginx章节,对其nginx介绍的非常详细,现把经常用到的Nginx配置参数中文说明摘录和nginx做负载均衡的本人真实演示实例抄录下来以便以后查看!

* **Nginx配置参数中文详细说明**

|  |  |  |
| --- | --- | --- |
| 001 | #定义Nginx运行的用户和用户组 | |
| 002 | user www www; |

|  |  |
| --- | --- |
| 003 | # |
| 004 | #nginx进程数,建议设置为等于CPU总核心数. | |

|  |  |  |
| --- | --- | --- |
| 005 | worker\_processes 8; | |
| 006 | # |

|  |  |  |
| --- | --- | --- |
| 007 | #全局错误日志定义类型,[ debug | info | notice | warn | error | crit ] | |
| 008 | error\_log /var/log/nginx/error.log info; |

|  |  |
| --- | --- |
| 009 | # |
| 010 | #进程文件 | |

|  |  |  |
| --- | --- | --- |
| 011 | pid /var/run/nginx.pid; | |
| 012 | # |

|  |  |  |
| --- | --- | --- |
| 013 | #一个nginx进程打开的最多文件描述符数目,理论值应该是最多打开文件数（系统的值ulimit -n）与nginx进程数相除,但是nginx分配请求并不均匀,所以建议与ulimit -n的值保持一致. | |
| 014 | worker\_rlimit\_nofile 65535; |

|  |  |
| --- | --- |
| 015 | # |
| 016 | #工作模式与连接数上限 | |

|  |  |  |
| --- | --- | --- |
| 017 | events | |
| 018 | { |

|  |  |  |
| --- | --- | --- |
| 019 | #参考事件模型,use [ kqueue | rtsig | epoll | /dev/poll | select | poll ]; epoll模型是Linux 2.6以上版本内核中的高性能网络I/O模型,如果跑在FreeBSD上面,就用kqueue模型. | |
| 020 | use epoll; |

|  |  |  |
| --- | --- | --- |
| 021 | #单个进程最大连接数（最大连接数=连接数\*进程数） | |
| 022 | worker\_connections 65535; |

|  |  |
| --- | --- |
| 023 | } |
| 024 | # |

|  |  |  |
| --- | --- | --- |
| 025 | #设定http服务器 | |
| 026 | http |

|  |  |
| --- | --- |
| 027 | { |
| 028 | include mime.types; #文件扩展名与文件类型映射表 | |

|  |  |  |
| --- | --- | --- |
| 029 | default\_type application/octet-stream; #默认文件类型 | |
| 030 | #charset utf-8; #默认编码 |

|  |  |  |
| --- | --- | --- |
| 031 | server\_names\_hash\_bucket\_size 128; #服务器名字的hash表大小 | |
| 032 | client\_header\_buffer\_size 32k; #上传文件大小限制 |

|  |  |  |
| --- | --- | --- |
| 033 | large\_client\_header\_buffers 4 64k; #设定请求缓 | |
| 034 | client\_max\_body\_size 8m; #设定请求缓 |

|  |  |
| --- | --- |
| 035 | # |
| 036 | #开启目录列表访问,合适下载服务器,默认关闭. | |

|  |  |
| --- | --- |
| 037 | autoindex on;                   #显示目录 |
| 038 | autoindex\_exact\_size on;        #显示文件大小 默认为on,显示出文件的确切大小,单位是bytes 改为off后,显示出文件的大概大小,单位是kB或者MB或者GB |

|  |  |  |
| --- | --- | --- |
| 039 | autoindex\_localtime on;         #显示文件时间 默认为off,显示的文件时间为GMT时间 改为on后,显示的文件时间为文件的服务器时间 | |
| 040 | # |

|  |  |  |
| --- | --- | --- |
| 041 | sendfile on; #开启高效文件传输模式,sendfile指令指定nginx是否调用sendfile函数来输出文件,对于普通应用设为 on,如果用来进行下载等应用磁盘IO重负载应用,可设置为off,以平衡磁盘与网络I/O处理速度,降低系统的负载.注意：如果图片显示不正常把这个改成off. | |
| 042 | tcp\_nopush on; #防止网络阻塞 |

|  |  |  |
| --- | --- | --- |
| 043 | tcp\_nodelay on; #防止网络阻塞 | |
| 044 | # |

|  |  |  |
| --- | --- | --- |
| 045 | keepalive\_timeout 120; #(单位s)设置客户端连接保持活动的超时时间,在超过这个时间后服务器会关闭该链接 | |
| 046 | # |

|  |  |  |
| --- | --- | --- |
| 047 | #FastCGI相关参数是为了改善网站的性能：减少资源占用,提高访问速度.下面参数看字面意思都能理解. | |
| 048 | fastcgi\_connect\_timeout 300; |

|  |  |
| --- | --- |
| 049 | fastcgi\_send\_timeout 300; |
| 050 | fastcgi\_read\_timeout 300; |

|  |  |  |
| --- | --- | --- |
| 051 | fastcgi\_buffer\_size 64k; | |
| 052 | fastcgi\_buffers 4 64k; |

|  |  |
| --- | --- |
| 053 | fastcgi\_busy\_buffers\_size 128k; |
| 054 | fastcgi\_temp\_file\_write\_size 128k; | |

|  |  |
| --- | --- |
| 055 | # |
| 056 | #gzip模块设置 | |

|  |  |
| --- | --- |
| 057 | gzip on; #开启gzip压缩输出 |
| 058 | gzip\_min\_length 1k; #允许压缩的页面的最小字节数,页面字节数从header偷得content-length中获取.默认是0,不管页面多大都进行压缩.建议设置成大于1k的字节数,小于1k可能会越压越大 | |

|  |  |
| --- | --- |
| 059 | gzip\_buffers 4 16k; #表示申请4个单位为16k的内存作为压缩结果流缓存,默认值是申请与原始数据大小相同的内存空间来存储gzip压缩结果 |
| 060 | gzip\_http\_version 1.1; #压缩版本（默认1.1,目前大部分浏览器已经支持gzip解压.前端如果是squid2.5请使用1.0） |

|  |  |
| --- | --- |
| 061 | gzip\_comp\_level 2; #压缩等级.1压缩比最小,处理速度快.9压缩比最大,比较消耗cpu资源,处理速度最慢,但是因为压缩比最大,所以包最小,传输速度快 |
| 062 | gzip\_types text/plain application/x-javascript text/css application/xml; |

|  |  |
| --- | --- |
| 063 | #压缩类型,默认就已经包含text/html,所以下面就不用再写了,写上去也不会有问题,但是会有一个warn. |
| 064 | gzip\_vary on;#选项可以让前端的缓存服务器缓存经过gzip压缩的页面.例如:用squid缓存经过nginx压缩的数据 |

|  |  |
| --- | --- |
| 065 | # |
| 066 | #开启限制IP连接数的时候需要使用 | |

|  |  |  |
| --- | --- | --- |
| 067 | #limit\_zone crawler $binary\_remote\_addr 10m; | |
| 068 | # |

|  |  |  |
| --- | --- | --- |
| 069 | ##upstream的负载均衡,四种调度算法(下例主讲)## | |
| 070 | # |

|  |  |  |
| --- | --- | --- |
| 071 | #虚拟主机的配置 | |
| 072 | server |

|  |  |
| --- | --- |
| 073 | { |
| 074 | #监听端口 | |

|  |  |
| --- | --- |
| 075 | listen 80; |
| 076 | #域名可以有多个,用空格隔开 | |

|  |  |
| --- | --- |
| 077 | server\_name wangying.sinaapp.com; |
| 078 | index index.html index.htm index.php; | |

|  |  |
| --- | --- |
| 079 | root /data/www/; |
| 080 | location ~ .\*\.(php|php5)?$ | |

|  |  |
| --- | --- |
| 081 | { |
| 082 | fastcgi\_pass 127.0.0.1:9000; | |

|  |  |  |
| --- | --- | --- |
| 083 | fastcgi\_index index.php; | |
| 084 | include fastcgi.conf; |

|  |  |
| --- | --- |
| 085 | } |
| 086 | #图片缓存时间设置 | |

|  |  |  |
| --- | --- | --- |
| 087 | location ~ .\*\.(gif|jpg|jpeg|png|bmp|swf)$ { | |
| 088 | expires 10d; |

|  |  |
| --- | --- |
| 089 | } |
| 090 | #JS和CSS缓存时间设置 | |

|  |  |  |
| --- | --- | --- |
| 091 | location ~ .\*\.(js|css)?$ { | |
| 092 | expires 1h; |

|  |  |  |
| --- | --- | --- |
| 093 | } | |
| 094 |  |

|  |  |
| --- | --- |
| 095 | #日志格式设定 |
| 096 | log\_format access '$remote\_addr - $remote\_user [$time\_local] "$request" ' | |

|  |  |
| --- | --- |
| 097 | '$status $body\_bytes\_sent "$http\_referer" ' |
| 098 | '"$http\_user\_agent" $http\_x\_forwarded\_for'; |

|  |  |
| --- | --- |
| 099 | #定义本虚拟主机的访问日志 |
| 100 | access\_log /var/log/nginx/access.log access; | |

|  |  |
| --- | --- |
| 101 | # |
| 102 | #设定查看Nginx状态的地址.StubStatus模块能够获取Nginx自上次启动以来的工作状态，此模块非核心模块，需要在Nginx编译安装时手工指定才能使用 | |

|  |  |  |
| --- | --- | --- |
| 103 | location /NginxStatus { | |
| 104 | stub\_status on; |

|  |  |
| --- | --- |
| 105 | access\_log on; |
| 106 | auth\_basic "NginxStatus"; | |

|  |  |
| --- | --- |
| 107 | auth\_basic\_user\_file conf/htpasswd; |
| 108 | #htpasswd文件的内容可以用apache提供的htpasswd工具来产生. | |

|  |  |  |
| --- | --- | --- |
| 109 | } | |
| 110 | } |

|  |  |
| --- | --- |
| 111 | } |

* **Nginx多台服务器实现负载均衡**

**Nginx负载均衡服务器：  
IP：192.168.1.1  
Web服务器列表：  
Web1:192.168.1.2  
Web2:192.168.1.3**  
实现目的：用户访问192.168.1.1服务器时，通过Nginx负载均衡到Web1和Web2服务器

|  |  |  |
| --- | --- | --- |
| 01 | http | |
| 02 | { |

|  |  |
| --- | --- |
| 03 | ##upstream的负载均衡,四种调度算法## |
| 04 | #调度算法1:轮询.每个请求按时间顺序逐一分配到不同的后端服务器,如果后端某台服务器宕机,故障系统被自动剔除,使用户访问不受影响 | |

|  |  |
| --- | --- |
| 05 | upstream webhost { |
| 06 | server 192.168.1.2:80 ; | |

|  |  |  |
| --- | --- | --- |
| 07 | server 192.168.1.3:80 ; | |
| 08 | } |

|  |  |  |
| --- | --- | --- |
| 09 | #调度算法2:weight(权重).可以根据机器配置定义权重.权重越高被分配到的几率越大 | |
| 10 | upstream webhost { |

|  |  |
| --- | --- |
| 11 | server 192.168.1.2:80 weight=2; |
| 12 | server 192.168.1.3:80 weight=3; |

|  |  |
| --- | --- |
| 13 | } |
| 14 | #调度算法3:ip\_hash. 每个请求按访问IP的hash结果分配,这样来自同一个IP的访客固定访问一个后端服务器,有效解决了动态网页存在的session共享问题 | |

|  |  |  |
| --- | --- | --- |
| 15 | upstream webhost { | |
| 16 | ip\_hash; |

|  |  |
| --- | --- |
| 17 | server 192.168.1.2:80 ; |
| 18 | server 192.168.1.3:80 ; |

|  |  |
| --- | --- |
| 19 | } |
| 20 | #调度算法4:url\_hash(需安装第三方插件).此方法按访问url的hash结果来分配请求,使每个url定向到同一个后端服务器,可以进一步提高后端缓存服务器的效率.Nginx本身是不支持url\_hash的,如果需要使用这种调度算法,必须安装Nginx 的hash软件包 | |

|  |  |
| --- | --- |
| 21 | upstream webhost { |
| 22 | server 192.168.1.2:80 ; | |

|  |  |  |
| --- | --- | --- |
| 23 | server 192.168.1.3:80 ; | |
| 24 | hash $request\_uri; |

|  |  |
| --- | --- |
| 25 | } |
| 26 | #调度算法5:fair(需安装第三方插件).这是比上面两个更加智能的负载均衡算法.此种算法可以依据页面大小和加载时间长短智能地进行负载均衡,也就是根据后端服务器的响应时间来分配请求,响应时间短的优先分配.Nginx本身是不支持fair的,如果需要使用这种调度算法,必须下载Nginx的upstream\_fair模块 | |

|  |  |
| --- | --- |
| 27 | # |
| 28 | #虚拟主机的配置(采用调度算法3:ip\_hash) | |

|  |  |  |
| --- | --- | --- |
| 29 | server | |
| 30 | { |

|  |  |
| --- | --- |
| 31 | listen  80; |
| 32 | server\_name  wangying.sinaapp.com; | |

|  |  |  |
| --- | --- | --- |
| 33 | #对 "/" 启用反向代理 | |
| 34 | location / { |

|  |  |  |
| --- | --- | --- |
| 35 | proxy\_pass http://webhost; | |
| 36 | proxy\_redirect off; |

|  |  |
| --- | --- |
| 37 | proxy\_set\_header X-Real-IP $remote\_addr; |
| 38 | #后端的Web服务器可以通过X-Forwarded-For获取用户真实IP |

|  |  |  |
| --- | --- | --- |
| 39 | proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for; | |
| 40 | #以下是一些反向代理的配置,可选. |

|  |  |
| --- | --- |
| 41 | proxy\_set\_header Host $host; |
| 42 | client\_max\_body\_size 10m; #允许客户端请求的最大单文件字节数 | |

|  |  |
| --- | --- |
| 43 | client\_body\_buffer\_size 128k; #缓冲区代理缓冲用户端请求的最大字节数, |
| 44 | proxy\_connect\_timeout 90; #nginx跟后端服务器连接超时时间(代理连接超时) |

|  |  |
| --- | --- |
| 45 | proxy\_send\_timeout 90; #后端服务器数据回传时间(代理发送超时) |
| 46 | proxy\_read\_timeout 90; #连接成功后,后端服务器响应时间(代理接收超时) |

|  |  |
| --- | --- |
| 47 | proxy\_buffer\_size 4k; #设置代理服务器（nginx）保存用户头信息的缓冲区大小 |
| 48 | proxy\_buffers 4 32k; #proxy\_buffers缓冲区,网页平均在32k以下的设置 |

|  |  |  |
| --- | --- | --- |
| 49 | proxy\_busy\_buffers\_size 64k; #高负荷下缓冲大小（proxy\_buffers\*2） | |
| 50 | proxy\_temp\_file\_write\_size 64k; |

|  |  |  |
| --- | --- | --- |
| 51 | #设定缓存文件夹大小,大于这个值,将从upstream服务器传 | |
| 52 | } |

|  |  |  |
| --- | --- | --- |
| 53 | } | |
| 54 | } |

**测试篇  
域名:wangying.sinaapp.com  
分别解析到192.168.1.1  
客户访问这三个站点的时候，Nginx根据客户访问的ip\_hash值，负载均衡到Web1和Web2服务器上  
虚拟主机的配置**

* **本地单台服务器实现动静分离多端口反向代理配置**

**Nginx负载均衡服务器：  
IP：192.168.1.1:80  
Web服务器(同台机器)列表：  
Web1:192.168.1.1:8080  
Web1:192.168.1.1:8081  
Web1:192.168.1.1:8082**  
实现目的：  
用户访问http://wangying.sinaapp.com，将其负载均衡到本地服务器的8080、8081、8082端口

[view source](http://wangying.sinaapp.com/archives/931#viewSource)

|  |  |  |
| --- | --- | --- |
| 01 | http | |
| 02 | { |

|  |  |  |
| --- | --- | --- |
| 03 | #因为服务器负载均衡到本地的8080、8081、8082端口,所以本地要增开8080,8081,8082端口作脚本解析 | |
| 04 | server { |

|  |  |
| --- | --- |
| 05 | listen       8080; |
| 06 | server\_name   wangying.sinaapp.com; | |

|  |  |  |
| --- | --- | --- |
| 07 | root  /mnt/hgfs/vmhtdocs/fastdfs/; | |
| 08 |  |

|  |  |
| --- | --- |
| 09 | location ~ \.php$ { |
| 10 | fastcgi\_pass   127.0.0.1:9000; | |

|  |  |  |
| --- | --- | --- |
| 11 | | fastcgi\_index  index.php; |
| 12 | fastcgi\_param  SCRIPT\_FILENAME  $document\_root$fastcgi\_script\_name; | | |

|  |  |  |
| --- | --- | --- |
| 13 | include        fastcgi\_params; | |
| 14 | } |

|  |  |  |
| --- | --- | --- |
| 15 | #由于下面80端口可知,8080,8081,8082只负责php动态程序解析,所以静态文件配置就不用设置了 | |
| 16 | } |

|  |  |
| --- | --- |
| 17 | server { |
| 18 | listen       8081; | |

|  |  |  |
| --- | --- | --- |
| 19 | server\_name   wangying.sinaapp.com; | |
| 20 | root  /mnt/hgfs/vmhtdocs/fastdfs/; |

|  |  |  |
| --- | --- | --- |
| 21 | index index.php index.html index.htm; | |
| 22 | location ~ \.php$ { |

|  |  |  |
| --- | --- | --- |
| 23 | fastcgi\_pass   127.0.0.1:9000; | |
| 24 | fastcgi\_index  index.php; |

|  |  |  |  |
| --- | --- | --- | --- |
| 25 | fastcgi\_param  SCRIPT\_FILENAME  $document\_root$fastcgi\_script\_name; | | |
| 26 | | include        fastcgi\_params; |

|  |  |  |
| --- | --- | --- |
| 27 | } | |
| 28 | } |

|  |  |  |
| --- | --- | --- |
| 29 | #8082的可仿照上面的server配置只是修改listen即可 | |
| 30 | # |

|  |  |
| --- | --- |
| 31 | #本地多端口负载均衡配置# |
| 32 | #因为是一台服务器,所以可以127.0.0.1代替其内网ip | |

|  |  |  |
| --- | --- | --- |
| 33 | #upstream 后面的主机名只是一个标识而已,可以是某个词语,也可以是域名,它与 proxy\_pass [http://webhost](http://webhost/) 相对应相同即可 | |
| 34 | upstream webhost { |

|  |  |
| --- | --- |
| 35 | server 127.0.0.1:8080; |
| 36 | server 127.0.0.1:8081; |

|  |  |  |
| --- | --- | --- |
| 37 | server 127.0.0.1:8082; | |
| 38 | } |

|  |  |  |
| --- | --- | --- |
| 39 | #本地80端口,接受请求做负载均衡 | |
| 40 | server |

|  |  |
| --- | --- |
| 41 | { |
| 42 | listen  80; | |

|  |  |  |
| --- | --- | --- |
| 43 | server\_name wangying.sinaapp.com; | |
| 44 | #本地动静分离反向代理配置 |

|  |  |  |
| --- | --- | --- |
| 45 | #所有php的页面均交由本地fastcgi处理 | |
| 46 | location ~ \.php$ { |

|  |  |
| --- | --- |
| 47 | proxy\_pass http://webhost; |
| 48 | proxy\_set\_header Host $host; | |

|  |  |
| --- | --- |
| 49 | proxy\_set\_header X-Real-IP $remote\_addr; |
| 50 | proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for; |

|  |  |
| --- | --- |
| 51 | } |
| 52 | #所有静态文件由nginx直接读取 | |

|  |  |
| --- | --- |
| 53 | #图片缓存时间设置 |
| 54 | location ~ .\*\.(gif|jpg|jpeg|png|bmp|swf)$ { | |

|  |  |  |
| --- | --- | --- |
| 55 | expires 10d; | |
| 56 | } |

|  |  |
| --- | --- |
| 57 | #JS和CSS缓存时间设置 |
| 58 | location ~ .\*\.(js|css)?$ { | |

|  |  |  |
| --- | --- | --- |
| 59 | expires 1h; | |
| 60 | } |

|  |  |
| --- | --- |
| 61 | } |