

nginx静态文件缓存的解决方案

nginx的一大功能就是完成静态资源的分离部署，减轻后端服务器的压力，如果给这些静态资源再加一级nginx的缓存，可以进一步提升访问效率。

第一步：添加nginx.conf的http级别的缓存配置

```
##cache##
proxy_connect_timeout 500;
#跟后端服务器连接的超时时间_发起握手等候响应超时时间
proxy_read_timeout 600;
#连接成功后_等候后端服务器响应的时间_其实已经进入后端的排队之中等候处理
proxy_send_timeout 500;
#后端服务器数据回传时间_就是在规定时间内后端服务器必须传完所有数据
proxy_buffer_size 128k;
#代理请求缓存区_这个缓存区间会保存用户的头信息以供Nginx进行规则处理_一般只要能保存下头信息即可
proxy_buffers 4 128k;
#同上 告诉Nginx保存单个用的几个Buffer最大用多大空间
proxy_busy_buffers_size 256k;
#如果系统很忙的时候可以申请更大的proxy_buffers 官方推荐*2
proxy_temp_file_write_size 128k;
#proxy缓存临时文件的大小
proxy_temp_path /usr/local/nginx/temp;
#用于指定本地目录来缓冲较大的代理请求
proxy_cache_path /usr/local/nginx/cache levels=1:2
keys_zone=cache_one:200m inactive=1d max_size=30g;
#设置web缓存区名为cache_one,内存缓存空间大小为12000M,自动清除超过15天没有被访问过的缓存数据,硬盘缓存空间大小200g
```

此处的重点在最后一句，缓存存储路径为：/usr/local/nginx/cache，levels=1:2代表缓存的目录结构为2级目录

如下图，缓存会在/usr/local/nginx/cache目录下生成，包含2级目录，在之下就是缓存文件，测试的时候可以到该目录下查看缓存文件是否生成。

```

[root@kaifa-202 ~]# cd /usr/local/nginx/
[root@kaifa-202 nginx]# ls
cache client_body_temp conf fastcgi_temp html htmlfile image logs proxy_temp sbin scgi_temp ssl temp uwsgi_temp
[root@kaifa-202 nginx]# cd cache/
[root@kaifa-202 cache]# ls
0 1 2 3 4 5 6 7 8 9 a b c d e f
[root@kaifa-202 cache]# pwd
/usr/local/nginx/cache
[root@kaifa-202 cache]# cd 0/
[root@kaifa-202 0]# ls
0b 0c 13 18 19 20 22 23 28 2e 35 39 4f 64 6e 74 80 8a 94 ac ad b0 b5 e3 e4 e8 f4 f8 fe
[root@kaifa-202 0]# cd 0b/
[root@kaifa-202 0b]# ls
9cad420a049c24ab76cdf829aaee40b0
[root@kaifa-202 0b]#

```

第二步：在访问静态文件的location上添加缓存

#静态数据保存时效

```

location ~ /\.html$ {
    proxy_pass http://source.qingk.cn;
    proxy_redirect off;
    proxy_cache cache_one;
    #此处的cache_one必须于上一步配置的缓存区域名称相同
    proxy_cache_valid 200 304 12h;
    proxy_cache_valid 301 302 1d;
    proxy_cache_valid any 1m;
    #不同的请求设置不同的缓存时效
    proxy_cache_key $uri$is_args$args;
    #生产缓存文件的key，通过4个string变量结合生成
    expires 30d;
    #其余类型的缓存时效为30天
    proxy_set_header X-Forwarded-Proto $scheme;
}

```

此处需要注意3点：

- 1、只有在proxy_pass的时候，才会生成缓存，下一次请求执行到proxy_pass的时候会判断是否有缓存，如果有则直接读缓存，返回给客户端，不会执行proxy_pass；如果没有，则执行proxy_pass，并按照规定生成缓存文件；可以到nginx的cache文件夹下看是否生成了缓存文件。
- 2、proxy_set_header Host \$host 这一句可能导致缓存失败，所以不能配置这一句。我在测试的时候遇到了这个问题，不明原理。
- 3、proxy_pass使用upstream出差，换成域名或ip则可行。

第三步：在proxy_pass跳转的location中配置静态文件的路径

```

location ~ .*\. (html)$ {

```

```
    default_type 'text/html';  
    root "/usr/local/openresty/nginx/html";  
}
```

将nginx本地存放静态文件的路径配到root指令处

如果没有这一句：default_type 'text/html'，所有的请求都默认是下载文件，而不是访问html页面

到此，静态文件缓存已经配置完成。但是还差很重要的最后一步，缓存生成之后会阻止访问进入后台和nginx本地，如果有更新，则更新内容无法生效，还需要一种手动清除缓存的机制。

第四步：清除缓存

缓存文件是根据proxy_cache_key这个指令生成的，所以找到对应的缓存文件，删除即可

```
location ~ /purge(/.*) {  
    #删除指定缓存区域cache_one的特定缓存文件$1$is_args$args  
    proxy_cache_purge cache_one $1$is_args$args;  
    #运行本机和10.0.217.0网段的机器访问，拒绝其它所有  
    allow          127.0.0.1;  
    allow          10.0.217.0/24;  
    deny           all;  
}
```

删除缓存用到proxy_cache_purge指令。

至此缓存生成和特定清除机制都已经实现。