（1）缓存介绍

1.代理服务器端缓存作用

减少后端压力，提高网站并发延时

2.缓存常见类型

服务器端缓存：代理缓存,获取服务器端内容进行缓存

浏览器端缓存

3.nginx代理缓存：proxy\_cache

（2）代理缓存配置

1.缓存配置

#vim /usr/local/nginx/conf/nginx.conf

upstream node {

server 192.9.191.31:8081;

server 192.9.191.31:8082;

}

proxy\_cache\_path /cache levels=1:2 keys\_zone=cache:10m max\_size=10g inactive=60m use\_temp\_path=off;

server {

listen 80;

server\_name www.test.com;

index index.html;

location / {

proxy\_pass http://node;

proxy\_cache cache;

proxy\_cache\_valid 200 304 12h;

proxy\_cache\_valid any 10m;

add\_header Nginx-Cache "$upstream\_cache\_status";

proxy\_next\_upstream error timeout invalid\_header http\_500 http\_502 http\_503 http\_504;

}

}

2.参数详解

proxy\_cache\_path /soft/cache levels=1:2 keys\_zone=cache:10m max\_size=10g inactive=60m use\_temp\_path=off;

#proxy\_cache //存放缓存临时文件

#levels //按照两层目录分级

#keys\_zone //开辟空间名,10m:开辟空间大小,1m可存放8000key

#max\_size //控制最大大小,超过后Nginx会启用淘汰规则

#inactive //60分钟没有被访问缓存会被清理

#use\_temp\_path //临时文件,会影响性能,建议关闭

proxy\_cache cache;

proxy\_cache\_valid 200 304 12h;

proxy\_cache\_valid any 10m;

add\_header Nginx-Cache "$upstream\_cache\_status";

proxy\_next\_upstream error timeout invalid\_header http\_500 http\_502 http\_503 http\_504;

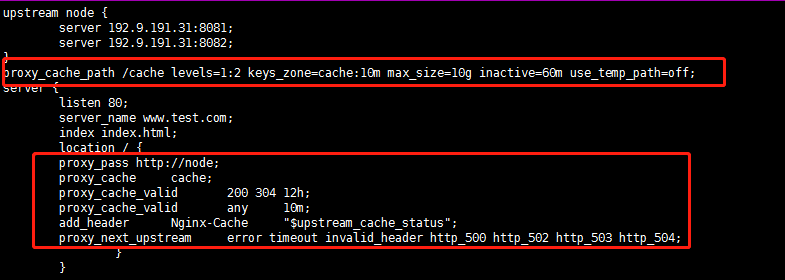
#proxy\_cache //开启缓存

#proxy\_cache\_valid //状态码200|304的过期为12h,其余状态码10分钟过期

#proxy\_cache\_key //缓存key

#add\_header //增加头信息,观察客户端respoce是否命中

#proxy\_next\_upstream //出现502-504或错误,会跳过此台服务器访问下一台服务器

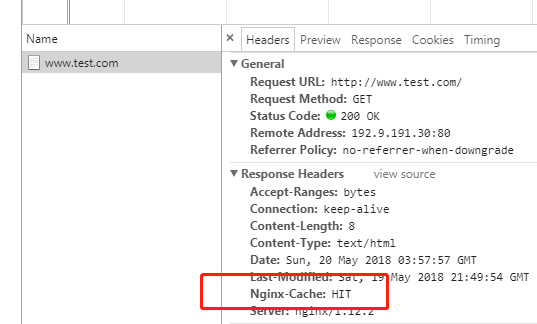


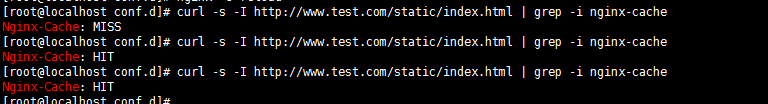
3.创建缓存目录

mkdir /cache

nginx -t

nginx -s reload





（3）清除缓存

1.rm删除已缓存的数据

rm -rf /cache/\*

2.通过ngx\_cache\_purge扩展模块清理,需要编译安装nginx

（4）部分页面不缓存

1.nginx配置

#vim /usr/local/nginx/conf/nginx.conf

upstream node {

server 192.9.191.31:8081;

server 192.9.191.31:8082;

}

proxy\_cache\_path /cache levels=1:2 keys\_zone=cache:10m max\_size=10g inactive=60m use\_temp\_path=off;

server {

listen 80;

server\_name www.test.com;

index index.html;

if ($request\_uri ~ ^/(static|login|register|password)) {

set $cookie\_nocache 1;

}

location / {

proxy\_pass http://node;

proxy\_cache cache;

proxy\_cache\_valid 200 304 12h;

proxy\_cache\_valid any 10m;

add\_header Nginx-Cache "$upstream\_cache\_status";

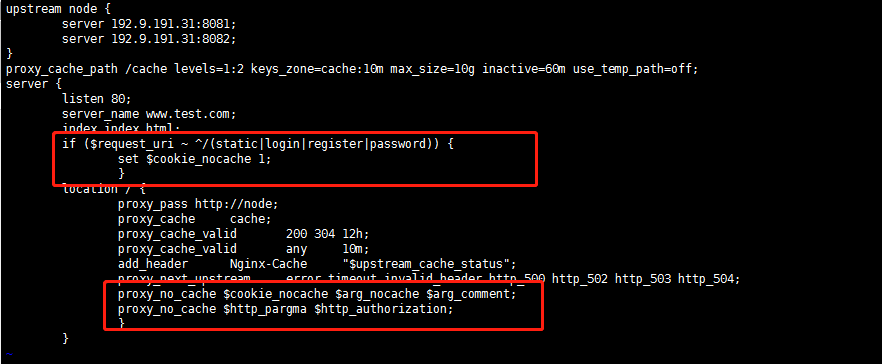
proxy\_next\_upstream error timeout invalid\_header http\_500 http\_502 http\_503 http\_504;

proxy\_no\_cache $cookie\_nocache $arg\_nocache $arg\_comment;

proxy\_no\_cache $http\_pargma $http\_authorization;

}

}

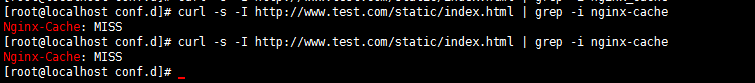


2.重启加验证

nginx -t

nginx -s reload

两次都没有命中



（4）统计日志命中率

1.日志格式:变量$upstream\_cache\_status"

#vim /usr/local/nginx/conf/nginx.conf

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for" "$upstream\_cache\_status"';

access\_log logs/access.log main;

error\_log logs/error.log;

3.统计日志命中率加入到计划任务中这里省略

awk '{if($NF = "HIT"){count++;}} END{printf "%.2f%",count/NR\*100}' /usr/local/nginx/logs/access.log