

【步骤总结，便于快速回顾】 查询优化技术之多级缓存



nginx的代理缓存配置方式

nginx proxy cache缓存的配置方法：在nginx.conf中申明如下内容

```
#申明一个cache缓存节点的路径
proxy_cache_path /usr/local/openresty/nginx/cache/temp levels=1:2 keys_zone=tmp_cache:100m inactive=7d max_size=

#//usr/local/openresty/nginx/cache/temp 把缓存文件放在哪里
#levels: 目录设置两层结构用来缓存
#keys_zone 指定了一个叫tmp_cache的缓存区，并且设置了100m的内存用来存储缓存key到文件路径的位置
#inactive 缓存文件超过7天后自动释放淘汰
#max_size 缓存文件总大小超过100g后自动释放淘汰
```

location内加入

```
proxy_cache tmp_cache;
proxy_cache_valid 200 206 304 302 10d;
proxy_cache_key $request_uri;
```

openresty扩展

加载lua模块

http模块下

```
lua_package_path "/usr/local/openresty/?.lua;;"; #lua 模块
lua_package_cpath "/usr/local/openresty/lualib/?.so;;"; #c模块
```

测试lua输出

```
location = /lua {
    default_type 'text/plain';
    content_by_lua 'ngx.say("hello, lua!")';
}
```

shared dict扩展

- 修改nginx.conf内，加入shared dictionary的扩展，声明128m的共享字典的访问内存

```
lua_shared_dict my_cache 128m;
```

- 设置location用来做访问shared dict的lua文件

```
location ^~/itemlua/get {
    default_type 'application/json';
    content_by_lua_file '/usr/local/openresty/nginx/lua/itemsharedic.lua';
}
```

- 然后使用itemlua url去访问验证

阿里云cdn缓存配置

- 生成域名
- 做cname绑定后启用cdn
- 验证对应资源，缓存策略和强制刷新策略