部署架构



对于/upload目录，URL需要包含规格参数，系统根据规格参数制造缩略图。对于/cms/giltstore的图片，用户根据原图的地址访问，访问到的是压缩过后的缩略图，规格为600宽度。如果图片本身宽度不到600，保持原有宽度。

1. 图片服务器篇

使得tengine支持lua脚本

* 安装luajit2.0.1

yum install openssl-devel

yum -y install pcre-devel

wget <http://luajit.org/download/LuaJIT-2.0.1.tar.gz>

tar zvxf LuaJIT-2.0.1.tar.gz

make

make install

* 安装tengine

wget <http://tengine.taobao.org/download/tengine-1.4.4.tar.gz>

tar zvxf tengine-1.4.4.tar.gz

wget https://github.com/simpl/ngx\_devel\_kit/archive/v0.2.18.tar.gz

tar zvxf ngx\_devel\_kit-0.2.18.tar.gz

wget <https://github.com/chaoslawful/lua-nginx-module/archive/v0.7.21.tar.gz>

tar zvxf lua-nginx-module-0.7.21.tar.gz

cd tengine-1.4.4

./configure --prefix=/usr/local/nginx --add-module=../ngx\_devel\_kit-0.2.18 --add-module=../lua-nginx-module-0.7.21 --with-http\_concat\_module --with-http\_realip\_module

make

make install

* 配置nginx

server\_name images.xiustatic.com,1.xiustatic.com,2.xiustatic.com,3.xiustatic.com,4.xiustatic.com,5.xiustatic.com,m.xiustatic.com,m1.xiustatic.com,m2.xiustatic.com,m3.xi

ustatic.com,m4.xiustatic.com,m5.xiustatic.com ;

upstream imgcluster {

server 172.16.3.53:9080 weight=4;

server 172.16.3.81:8083 weight=1;

ip\_hash;

keepalive 512;

check interval=3000 rise=2 fall=5 timeout=1000 type=http;

check\_http\_send "GET /?heartbeatcheck=1 HTTP/1.0\r\n\r\n";

check\_http\_expect\_alive http\_2xx http\_3xx;

}

#upload request handling

location /cms/giltstore/ {

set $reqPath "";

set $pathTemp "";

if ($uri ~\* "/cms/giltstore/(.\*)[.](jpg|gif|png)([?].\*)?$") {

set $reqPath "/cms/giltstore/$1\_600\_0.$2";

set $pathTemp "/cms/giltstore/$1.$2";

}

set $reqPath1 "";

if ($uri ~\* "/cms/giltstore/(.\*)") {

set $reqPath1 "/cms/giltstore/$1";

}

set $ismobilerequest "0";

if ($host ~\* "zoshow[.]com") {

set $ismobilerequest "1";

}

if ($ismobilerequest = "0") {

set $reqPath $pathTemp;

}

rewrite\_by\_lua\_file /www/target/nginx/router/ImageLocator.lua;

root /www/image\_server/xiu2.0/public/;

}

#upload request handling

location /upload/ {

set $reqPath "";

if ($uri ~\* "/upload/(.\*[.]jpg|[.]gif|[.]png)([?].\*)?$") {

set $reqPath "/upload/$1$2";

}

set $reqPath1 "";

if ($uri ~\* "/upload/(.\*)") {

set $reqPath1 "/upload/$1";

}

rewrite\_by\_lua\_file /www/target/nginx/router/ImageLocator.lua;

root /www/image\_server/xiu2.0/public/;

}

location /internal\_images/ {

allow all;

alias /www/image\_server/xiu2.0/public/;

expires 2h;

#proxy\_intercept\_errors on;

#if (!-f $request\_filename) {

# proxy\_pass http://172.16.1.3;

# break;

#}

}

location /internal\_thumbnail/ {

allow all;

alias /www/image\_server/xiu2.0/public/thumbnail/;

expires 2h;

}

location /upload\_thumbnail/ {

proxy\_pass http://imgcluster;

}

上传ImageLocator.lua到conf目录

修改该文件配置目录

--

--原图目录

local ORIGINAL\_IMAGE\_ROOT = "/www/images/upload"

--缩略图目录

local THUMBNAIL\_IMAEG\_ROOT = "/www/images/thumbnail"

1. 缩略图生成服务器篇
2. 安装必备软件以及相关依赖库

* 安装 graphicsmagick
  + yum install libjpeg libjpeg-devel libpng libpng-devel

cd GraphicsMagick-1.3.18

./configure --enable-shared --disable-openmp

make

make install

* + 验证安装

gm convert /www/target/images/ori1.jpg -thumbnail 400x400 -gravity center -quality 90.0 -extent 400x400 -background white /www/target/thumbnail/ori1\_400\_400.jpg

* 安装 luajit2.0.1

yum install openssl-devel

yum -y install pcre-devel

wget <http://luajit.org/download/LuaJIT-2.0.1.tar.gz>

tar zvxf LuaJIT-2.0.1.tar.gz

make

make install

* 安装tengine + lua + nginx\_devel\_kit
  + yum install openssl-devel
  + yum -y install pcre-devel
  + 修改/etc/profile

export LD\_LIBRARY\_PATH=/usr/local/lib/:$LD\_LIBRARY\_PATH

* + 编译tengine

./configure

--prefix=/usr/local/nginx

--add-module=../ngx\_devel\_kit-0.2.18

--add-module=../lua-nginx-module-0.7.21

* 升级cmake到2.8

./configure

make

make install

vi /etc/profile

export CMAKE\_ROOT=/usr/local/share/cmake-2.8

* 安装git

Yum install git

* 安装 luarocks 并通过luarocks安装相关依赖的lua库

wget curl,git三个工具要提前安装。如果没有安装，可以通过yum安装:

yum install wget

yum install curl

yum install git

大部分情况下linux已经安装了lua（不是luajit），如果没有需要自行安装

Yum install readline

Yum install readline-devel

Cd lua-5.1.4

Make linux

Make install

执行lua，应该肯到命令提示行，此程序安装与/usr/local/bin目录。

现在开始安装luarocks

cd luarocks-2.0

./configure --with-lua-include=/usr/local/include/luajit-2.0

./make

* 修改/etc/profile

export LD\_LIBRARY\_PATH=/usr/local/lib/:$LD\_LIBRARY\_PATH

export CMAKE\_ROOT=/usr/local/share/cmake-2.8/

export PKG\_CONFIG\_PATH=/usr/local/lib/pkgconfig

* + 安装torch7

cd torch-7.0-0

luarocks make

* + luarocks --from=http://www.torch.ch/resources/rocks/ install sys
  + luarocks --from=http://www.torch.ch/resources/rocks/ install image
  + luarocks --from=http://www.torch.ch/resources/rocks/ install torchffi
  + 安装graphicsmagick lua模块

因graphicsmagick已经打了补丁，网上下载是不行的

cd graphicsmagick-master

luarocks make

安装完毕后，通过luarocks list命令，应该可以看到以下已安装模块

image

1.0-0 (installed) - /usr/local/lib/luarocks/rocks

sys

1.0-0 (installed) - /usr/local/lib/luarocks/rocks

torch

7.0-0 (installed) - /usr/local/lib/luarocks/rocks

torchffi

1.0-0 (installed) - /usr/local/lib/luarocks/rocks

xlua

1.0-0 (installed) - /usr/local/lib/luarocks/rocks

* 安装zeromq

cd /www/target/tengine/zeromq-3.2.3

./configure

make

make install

* 安装lua-zmq

cd lua-zmq

mkdir build

cd build

cmake .. -DLUA\_INCLUDE\_DIR=/usr/local/include/luajit-2.0

make

make install

1. 配置服务器

* 覆盖原有的nginx.conf，修改nginx.conf的/www/images/public 为真是的图片存储目录，此目录和路由器指向的图片服务器目录必须是同一个。
* 上传ImageProcessor.lua到/usr/local/share/lua/5.1目录
* 上传thumbnail.lua到服务器的/usr/local/nginx/conf目录
* 修改thumbnail.lua

找到

originalImagesRoot = "/www/images/public/upload/",

thumbnailImagesRoot = "/www/images/public/thumbnail/",

按照实际的图片存储目录配置

修改serverUrl = "tcp://172.16.3.28:5559"，172.16.3.28需要指向图片压缩服务器的ip(请参考下一章节)

1. 图片压缩服务器篇

* Mount图片目录到图片存储，保证和缩略图的mount路径一致
* 安装pagespeed

cd page-speed-1.9

make

cd out/Release/

chmod +x optimize\_image\_bin

* 安装zeromq

cd /www/target/tengine/zeromq-3.2.3

./configure

make

make install

* 安装lua-zmq

cd lua-zmq

mkdir build

cd build

cmake .. -DLUA\_INCLUDE\_DIR=/usr/local/include/luajit-2.0

make

make install

--cp /usr/local/lib/lua/5.1/zmq.so /usr/lib64/lua/5.1/

--cp -rf /usr/local/share/lua/5.1/zmq /usr/share/lua/5.1/

* 安装压缩程序

mkdir compressimage

cd compressimage

端口指定为5559，5560

mkdir broker

cd broker

rz msg-router/rrbroker.lua

rz msg-router/zhelpers.lua

rz json.lua

nohup luajit rrbroker.lua & broker.out &

端口指定为5560，5561

cd compressimage

mkdir compress

cd compress

rz compressimage/zhelpers.lua

rz compressimage/compressimage.lua

rz json.lua

compressimage.lua配置

设置pagespeed程序optimize\_image\_bin的位置

nohup luajit compressimage.lua

1. 原图修改删除监控程序

缩略图在原图修改或者删除之后，需要作废。当缩略图生成之后，监控程序会根据跟踪原图的修改时间，如果修改时间发生了变化或者原图已经删除，系统将会删除此由次原图生成的缩略图。

首先部署消息接收路由器。

mkdir broker

cd broker

rz msg-router/rrbroker.lua

rz msg-router/zhelpers.lua

rz json.lua

nohup luajit rrbroker.lua & broker.out &

端口为5561，5562

上传image-monitor.zip并且解压，修改config/config.properties，指定mongodb的访问url以及图片的存储目录等。

dbUrl=192.168.3.109:27017

serverUrl=tcp://localhost:5562

original-image-home=/www/images/public/upload

thumbnail-home=/www/images/public/thumbnail

checkInterval=3000

1. FAQ

* /usr/local/nginx/sbin/nginx: error while loading shared libraries: libpcre.so.0: or directory

ln -s /usr/local/lib/libpcre.so.0 /usr/lib/libpcre.so.0

* None of the required 'libzmq' found

export PKG\_CONFIG\_PATH=/usr/local/lib/pkgconfig

* error loading module 'zmq' from file '/usr/lib64/lua/5.1/zmq.so':

copy zmq.so到package.cpath

* libzmq.so.3: cannot open shared object file: No such file or directory

export LD\_LIBRARY\_PATH=/usr/local/lib/:$LD\_LIBRARY\_PATH

FAQ：

Gm convert 失败，原因是libjpeg库，libpng库没有安装。

通过gm convert成功，但是通过lua读不了图片，原因是-enable-shared参数没有设置。

./configure '--enable-shared' '--disable-openmp '

性能很低 --disable-openmp 编译参数没有设置。

yum install -y gcc gcc-c++