Nginx关键技术分析及运用—爱旅行项目

## Nginx简介

1. 正向代理

客户端通过代理服务器访问原始服务器。对于客户端而言，正向代理所代理的是客户端，那么客户端必须要进行一些特别的设置。代理流程如下图1所示：

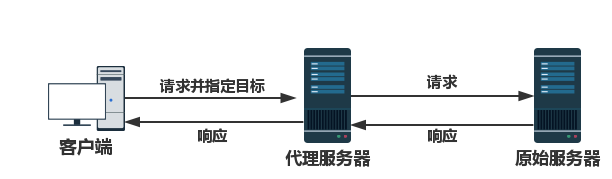


图1

1. 反向代理

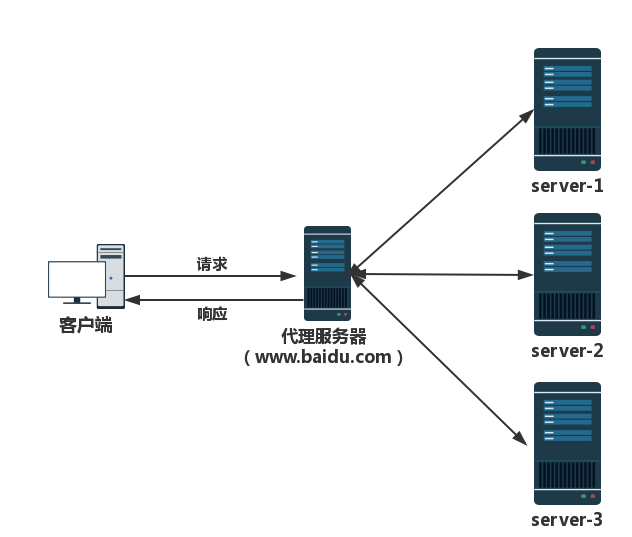
客户端发送请求到代理服务器，然后代理服务器将请求转发给内部网络上的其他服务器（原始服务器），并将从原始服务器上得到的结果返回给客户端，此时代理服务器就是代理的服务端，客户端无须进行特别的设置，对外就表现为一个反向代理服务器。代理流程如下图2所示：

图2

1. 负载均衡(Load Balance)

负载均衡是指将请求/数据【均匀】分摊到多个操作单元上执行，负载均衡的关键在于均匀，均衡的分摊压力。下图3是一个简单负载均衡集群。我们的爱旅行项目就是采用这种集群架构来实现负载均衡的。

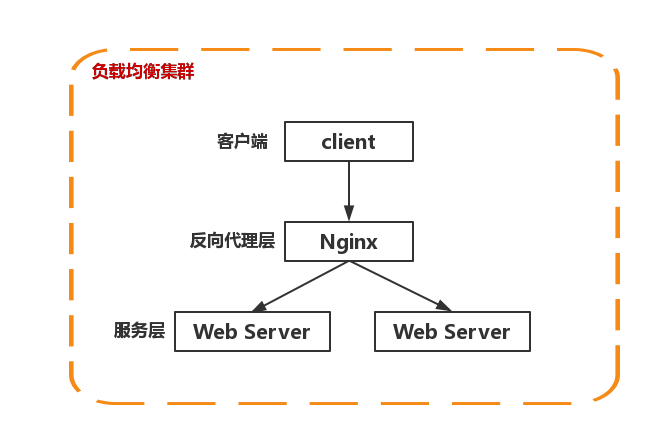


图3

1. Nginx

Nginx (engine x) 是一个很强大的高性能的web服务器、[反向代理](http://baike.baidu.com/item/%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86)服务器。并且作为反向代理服务器实现了负载均衡。目前国内使用Nginx 网站有：百度、[京东](http://baike.baidu.com/item/%E4%BA%AC%E4%B8%9C)、[新浪](http://baike.baidu.com/item/%E6%96%B0%E6%B5%AA)、[网易](http://baike.baidu.com/item/%E7%BD%91%E6%98%93)、[腾讯](http://baike.baidu.com/item/%E8%85%BE%E8%AE%AF)、[淘宝](http://baike.baidu.com/item/%E6%B7%98%E5%AE%9D)（Tengine）等。

## 1.2 安装配置

### 1.2.1 环境准备

1. Nginx安装包
2. Widows版
3. Linux版

安装包：nginx-1.12.0.tar.gz

下载地址： http://nginx.org/en/download.html

1. 服务器环境
2. CentOS 6.4 64位
3. 安装c++编译环境（ yum install gcc-c++）
4. 下载地址： GraphicsMagick

### 1.2.2 安装配置步骤

1. 安装模块依赖库
2. pcre库（rewrite）

yum install pcre\* 或者 pcre-8.32.tar.gz

下载地址： http://www.pcre.org/

1. openssl库（ssl）

yum install openssl\* 或者openssl-fips-2.0.16.tar.gz

下载地址： http://www.openssl.org/

1. zlib库（gzip）

yum install zlib\*或者zlib-1.2.11.tar.gz

下载地址：http://www.zlib.net/

1. 安装Nginx

./configure

make&make install

1. 开放80端口

vim /etc/sysconfig/iptables

service iptables restart  #最后重启防火墙使配置生效

1. 启动Nginx

命令：usr/local/nginx/sbin/nginx

1. 访问Nginx

http://服务器IP

### 1.2.3 常用命令

1. 负责裁剪的Lua脚本（/usr/local/Tengine/lua/ImageResizer.lua）
2. 启动：usr/local/nginx/sbin/nginx
3. 停止：usr/local/nginx/sbin/nginx -s stop
4. 重启：usr/local/nginx/sbin/nginx –s reload
5. 检查配置文件(nginx.conf)是否合法： usr/local/nginx/sbin/nginx –t

### 1.2.4 Nginx配置文件

Nginx有一个很重要的配置文件：conf目录下nginx.conf文件，Nginx服务器的基础配置，默认的配置都存放于这个文件内。若要实现反向代理，负载均衡，均需要在该文件内进行配置。我们先来看下该配置文件的结构：

1. 全局块

配置影响Nginx全局的指令。一般有运行Nginx服务器的用户组，Nginx进程pid存放路径，日志存放路径，配置文件引入，允许生成worker process数等。

1. events块

配置影响Nginx服务器或与用户的网络连接。有每个进程的最大连接数，选取哪种事件驱动模型处理连接请求，是否允许同时接受多个网路连接，开启多个网络连接序列化等。

1. http块

设定http服务器，利用它的反向代理功能提供负载均衡支持，可以嵌套多个server，如果http服务，支持了多个虚拟主机，那么在http上下文里，就会出现多个server上下文。里面包括配置反向代理功能提供负载均衡支持，缓存，日志定义等绝大多数功能和第三方模块的配置。

1. server块

配置虚拟主机的相关参数，一个http中可以有多个server。

1. location块

配置请求的路由，以及各种页面的处理情况，location根据其后面的正则进行匹配，对请求URL过滤。

## 1.3 项目实现负载均衡

1. 反向代理

Nginx只处理静态不处理动态内容，动态内容交给后台Tomcat处理。关键代码片段如下：

|  |
| --- |
| upstream itripbiz\_server  {  server 127.0.0.1:8080;  }  server {  listen 80;  server\_name itrip.project.bdqn.cn;  root /data/itrip/itripfront; #前端静态工程  location / {  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_pass http://itripbiz\_server; #反向代理地址  } |

1. 实现负载均衡
2. 轮询

|  |
| --- |
| upstream itripbiz\_server  {  server 127.0.0.1:8080;  server 127.0.0.1:8082;  }  server {  listen 80;  server\_name itrip.project.bdqn.cn;  root /data/itrip/itripfront; #前端静态工程  location / {  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_pass http://itripbiz\_server; #反向代理地址  } |

1. 热备（backup）

|  |
| --- |
| upstream itripbiz\_server  {  server 127.0.0.1:8080;  server 127.0.0.1:8082 backup;  }  server {  listen 80;  server\_name itrip.project.bdqn.cn;  root /data/itrip/itripfront; #前端静态工程  location / {  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_pass http://itripbiz\_server; #反向代理地址  } |

1. 权重（weight）

|  |
| --- |
| upstream itripbiz\_server  {  server 127.0.0.1:8080;  server 127.0.0.1:8082 weight=2;  }  server {  listen 80;  server\_name itrip.project.bdqn.cn;  root /data/itrip/itripfront; #前端静态工程  location / {  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_pass http://itripbiz\_server; #反向代理地址  } |

1. IP地址hash（ip\_hash）

可解决Tomcat之间session共享问题。并且基于权重的负载均衡和基于IP地址哈希的负载均衡可以组合在一起使用。

|  |
| --- |
| upstream itripbiz\_server  {  ip\_hash  server 127.0.0.1:8080;  server 127.0.0.1:8082;  }  server {  listen 80;  server\_name itrip.project.bdqn.cn;  root /data/itrip/itripfront; #前端静态工程  location / {  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_pass http://itripbiz\_server; #反向代理地址  } |