Nginx，主讲：汤小洋

一、Nginx简介

1. Nginx是什么？

Nginx (engine x) 是一个高性能的HTTP和反向代理服务器，也是一个IMAP/POP3/SMTP服务器

Nginx可以作为一个Web服务器进行网站的发布，也可以作为反向代理服务器进行负载均衡的实现

常见的Web服务器：Tomcat、Apache、Nginx、Weblogic等

2. 特点

占用内存少、并发能力强

二、搭建Nginx环境

1.安装nginx

两种方式：

- 源代码安装：需要编译 ./configure ——> make ——> make install
- 在线安装，参考：http://nginx.org/en/linux_packages.html

在线安装：

1. 下载nginx认证文件，并添加到apt-key中

```
sudo apt-key add nginx_signing.key
```

2. 配置apt源，添加nginx软件源

```
sudo vi /etc/apt/sources.list
    deb http://nginx.org/packages/ubuntu/ trusty nginx
    deb-src http://nginx.org/packages/ubuntu/ trusty nginx
```

3. 更新apt软件源，并安装nginx

```
sudo apt-get update
sudo apt-get install nginx
```

4. 访问测试

   http://ip地址

   注：Nginx默认使用的是80端口

2. 目录结构

执行 `whereis nginx` 查看

命令程序：

- /usr/sbin/nginx

配置文件：

- /etc/nginx/nginx.conf

日志目录：

- /var/log/nginx/

默认虚拟主机目录：

- /usr/share/nginx/html

3. 相关命令

```
netstat -ntpl | grep 80   #查看进程信息
ps aux | grep ntinx

sudo nginx  #启动
sudo nginx -s stop  #停止
sudo nginx -s reload  #重启
sudo nginx -c /etc/nginx/nginx.conf  #使用指定的配置文件启动
sudo nginx -t # 测试配置文件是否有错误
sudo nginx -v  #查看版本信息
```

4. 关于配置文件

主配置文件nginx.conf，包含三部分内容：全局配置、工作模式配置、HTTP配置

```
#运行nginx的用户
user  nginx;
#工作进程的数量，可以根据CPU的核心总数来设置
worker_processes  4;

#错误日志文件的位置及输出级别
error_log  /var/log/nginx/error.log warn;
#PID文件的位置
pid        /var/run/nginx.pid;

#工作模式配置
events {
    #每个进程最大处理的连接数
    worker_connections  10000;
}

#HTTP配置
http {
    #支持的媒体类型
    include       /etc/nginx/mime.types;
    #默认的类型
    default_type  application/octet-stream;

    #日志格式
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    #访问日志文件的位置
    access_log  /var/log/nginx/access.log  main;

    #是否调用sendfile函数来输出文件
    sendfile        on;
    #tcp_nopush     on;

    #连接超时时间
    keepalive_timeout  65;

    #开启gzip压缩
    #gzip  on;

    #引入外部配置文件，包含虚拟主机的配置
    include /etc/nginx/conf.d/*.conf;
}
```

虚拟主机配置文件/etc/nginx/conf.d/default.conf，可以定义多个虚拟主机配置文件

```
#虚拟主机的配置
server {
    #监听端口
    listen       80;
    #服务器域名
    server_name  localhost;

    #网页的默认编码
    #charset koi8-r;
    #访问该虚拟主机的日志位置
    #access_log  /var/log/nginx/host.access.log  main;

    #根据目录配置
    location / {
        #网站根目录的配置
        root   /usr/share/nginx/html;
        #默认首页
        index  index.html index.htm;
    }

    #错误的反馈页面
    error_page   500 502 503 504  /50x.html;
    #错误页面的配置
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}
```

将原配置文件备份，养成数据备份的习惯

三、HTTP服务器

1. 简介

虚拟主机：把一台物理服务器划分为多个虚拟的服务器，称为虚拟主机

每个虚拟主机对应一个Web站点，其实就是在一台服务器上搭建多个网站

2. 虚拟主机

步骤：

1. 准备网站目录及测试页面

```
mkdir www
cd www
mkdir ums
mkdir sms
echo welcome to ums > ums/index.html
echo welcome to sms > sms/index.html
```

2. 创建虚拟主机配置文件并配置

```
cd /etc/nginx/conf.d/
sudo cp default.conf ums.conf
sudo cp default.conf sms.conf
sudo vi ums.conf
    server {
      listen       80;
      server_name  www.ums.com;

      location / {
          root   /home/soft01/www/ums;
          index  index.html index.htm;
      }
  }
sudo vi sms.conf
    server {
      listen       80;
      server_name  www.sms.com;

      location / {
          root   /home/soft01/www/sms;
          index  index.html index.htm;
      }
  }
```

实现虚拟主机的三种方式：

- 基于不同的IP
- 基于不同的端口
- 基于不同的域名

3. 配置域名解析

在客户端主机中配置域名解析

windows：C:\Windows\System32\drivers\etc\hosts

Linux/Mac：/etc/hosts

```
192.168.1.59 www.ums.com
192.168.1.59 www.sms.com
```

域名解析的过程：查找hosts文件——>DNS

注：该方式仅是本地测试时使用的，实际应用中要购买注册域名

3. 作为图片服务器

使用Nginx作为图片服务器：

- 上传：使用ftp或sftp上传图片到服务器指定的ftp目录下
- 下载：通过访问Nginx服务器来访问ftp目录下的图片文件，即使用HTTP请求来访问资源文件，而不是通过FTP请求

步骤：

1. 创建存放图片的文件夹，并上传图片到该目录中

```
mkdir /home/soft01/www/images
```

2. 配置Nginx

```
sudo vi /etc/nginx/conf.d/default.conf
     location /images {
        root   /home/soft01/www;
        autoindex on; #打开目录浏览功能
    }
```

访问url：http://ip/images 实际物理路径：/home/soft01/www/images

3. 访问

http://ip/images/xxx.img

/home/soft01

四、反向代理

1. 简介

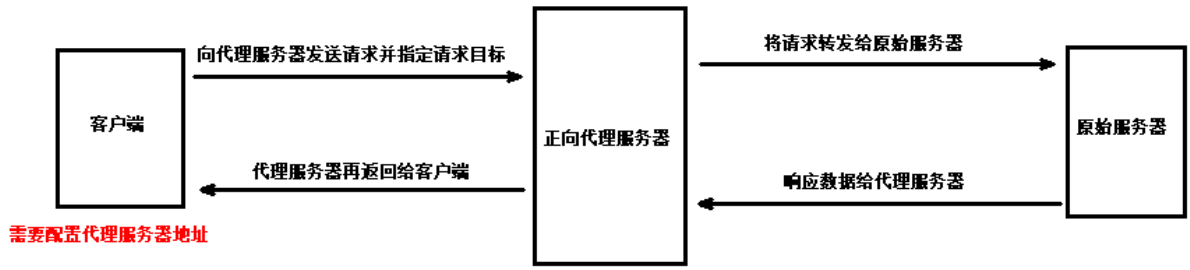
1. 正常请求

客户端发送请求到服务器，服务器接收请求并响应数据

2. 正向代理

概念：位于客户端和原始服务器之间的服务器，为了从原始服务器获取数据，客户端向代理服务器发送请求并指定请求目标(原始服务器)，然后代理服务器将请求转换给原始服务器，并将响应的数据返回给客户端

正向代理是客户端使用的，对客户端进行代理，客户端知道并主动使用代理



作用：

- 访问原来无法访问的资源（google、fackbook等），翻墙
- 可以做缓存，加速资源的访问
- 对客户端上网进行认证授权
- 上网行为管理，记录用户访问记录，对外隐藏用户信息

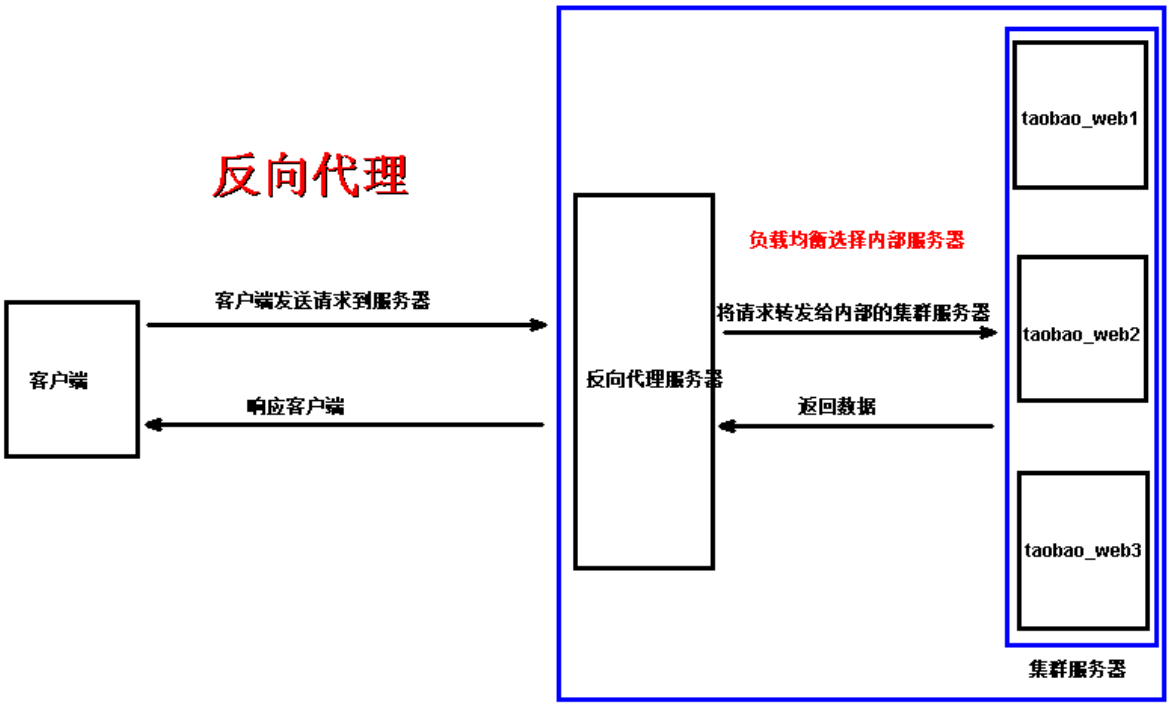
3. 反向代理

概念：客户端发送请求到服务器（客户端认为是原始服务器，实际上是一台反向服务器），反向代理服务器接收请求并将请求转发给内部网络中的多台集群服务器，并将响应的数据返回给客户端

反向代理一般用于服务器集群、分布式等，实现负载均衡

作用：

- 负载均衡，提高处理和响应速度
- 保证内网的安全，隐藏服务器信息，防止Web攻击



2. 配置

使用Nginx作为其他服务器

步骤：

1. 创建虚拟主机配置文件，并配置反向代理

```
scd /etc/nginx/conf.d
sudo cp default.conf proxy.conf
sudo vi proxy.conf
    #后台服务器列表
  upstream tomcat_server{
      server 192.168.1.66:8080;
  }

  server {
      listen       80;
      server_name  www.tomcat.com;

      location / {
          proxy_pass http://tomcat_server;  #指定代理的后台服务器
      }
  }
```

2. 配置域名解析
3. 访问

   http://tomcat.com

五、负载均衡

1. 简介

概念：将接收到的请求按照一定的规则分发到不同的服务器进行处理，从而提高系统响应和处理速度，称为负载均衡

2. 配置

步骤：

1. 准备网站（模拟淘宝，后面有多台服务器）

```
#拷贝两个tomcat
cp -r apache-tomcat-8.5.30 taobao1
cp -r apache-tomcat-8.5.30 taobao2
#修改tomcat端口
vi taobao1/conf/server.xml
vi taobao2/conf/server.xml
#修改页面
vi taobao1/webapps/ROOT/index.jsp
vi taobao2/webapps/ROOT/index.jsp
#启动tomcat
./startup.sh
```

2. 创建虚拟主机配置文件，并配置负载均衡

```
sudo cp proxy.conf taobao.conf
sudo vi taobao.conf
    #后台服务器列表
  upstream taobao_server{
      server 192.168.1.66:8081 weight=3; #weight表示权重，权重越高被分配到的几率越大
      server 192.168.1.66:8082 weight=7;
  }

  server {
      listen       80;
      server_name  www.taobao.com;

      location / {
          proxy_pass http://taobao_server;  #指定代理的后台服务器
      }
  }
```

六、动静分离

1. 简介

问题：tomcat在处理静态资源时效率不高，默认情况下所有资源都由tomcat处理，会导致Web应用响应慢，占用系统资源

解决：将静态资源交由Nginx处理，动态资源仍由tomcat处理，实现动静分离

实际上就是把Nginx作为静态资源服务器

2. 配置

步骤：

1. 编辑taobao.conf，配置动态分离

```
sudo vi /etc/nginx/conf.d/taobao.conf
    #处理静态资源
    location ~ .*\.(js|css|ico|png|jpg|eot|svg|ttf|woff) {
        root /home/soft01/www/static;
    }
```

2. 创建存放静态资源的文件夹，并将资源资源放到该目录中

```
cd /home/soft01/www
mkdir static
cd static
chmod 777 *
cd /home/soft01/software/taobao1/webapp/ROOT
cp tomcat.css tomcat.png  /home/soft01/www/static
```

高并发的处理：

- 负载均衡：集群
- 动静分离：使用Nginx、CDN
- 缓存：以空间换时间，提高系统效率
- 限流：流量控制
- 降级：服务降载