

Nginx集群环境搭建

Ningx集群环境搭建

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 代理服务器。一直纳闷这个X是怎么来的在网上查了查原来X代表很牛逼的样子，Nginx就是代表一个非常牛逼的引擎服务器系统，在很多年以前记得都是用的apache来作为分流以及负载均衡的作用，在过了几年中一个俄罗斯人又开发了一个性能更高的服务器来处理客户端向浏览器发出的各种请求，它从技术角度解决了大数量高并发的用户量访问，官网上面写的并发数量可以达到5万，实际项目应用中可能也打不到这么多，如果可以达到2、3万并发访问量，也是一个用户非常多的网站了。

Ningx集群环境搭建

Nginx与Apache服务器对比

- 1.轻量级，同样起web 服务，比apache 占用更少的内存及资源 抗并发，nginx 处理请求是异步非阻塞的，而apache 则是阻塞型的，在高并发下nginx 能保持低资源低消耗高性能 。nginx运行起来占用的资源CUP非常少，对于启动、重启都非常快速，在低配电脑上也可以快速的运行。
- 2.Nginx 配置简洁,Apache 复杂
对于这一点使用过nginx的用户来说就会深有体会，如果不是学习他的原理以及各种细节，几乎一会的时间就可以搭起来一个nginx集群服务器，搭建可以按着下面的步骤自己搭建。
- 3.最核心的区别在于apache是同步多进程模型，一个连接对应一个进程； nginx是异步的，多个连接（万级别）可以对应一个进程。
- 4.nginx的负载能力比apache高很多

Ningx集群环境搭建

典型案例

中国大陆使用nginx网站用户有：新浪、网易、腾讯等。这些大的公司都在使用nginx作为网站的负载均衡。可以说，它已经是一个比较成熟的框架，值得我们去学习和了解，通过学习和了解这个服务,也了解了一些多线程、阻塞、非阻塞的概念，多线程是一个很不错的提升性能的解决方案，但有时在多个线程之间切换以及创建关闭线程很消耗系统资源，接触的一些多线程模式可以解决多线程带来的并发访问问题，如果master/worker,fulture，生产者/消费者模式等等，在nginx里面就用到了一个多线程模式，master/worker模式，这个模式在nginx中起着重要的作用。

Ningx集群环境搭建

安装并启动Nginx

1 安装

```
#yum -y install nginx
```

2 关闭防火墙

Ningx集群环境搭建

启动Nginx

`#systemctl start nginx`

`#systemctl enable nginx`

在客户端浏览器验证测试

Ningx集群环境搭建

❖ Nginx初始页面



Ningx集群环境搭建

Nginx服务不间断重启(平滑重启)

#kill -HUP `cat /run/nginx.pod`

其中 **HUP**信号代表的为重新加载配置,即关闭原有进程并开启新进程

Ningx集群环境搭建

制作虚拟主机

虚拟主机配置文件：

```
# vim /etc/nginx/conf.d/niliu.host.conf

server {

listen      80 ;                //定义监听端口

server_name  www.niliu.edu;    //定义主机的FQDN

location / {

index index.html;              //定义索引文件

root  /usr/share/nginx/html;   // 定义内容根目录

}

}
```

Ningx集群环境搭建

制作虚拟主机

```
#cd /usr/share/nginx/html
```

```
#mv index.html index.html.bf
```

```
#echo "test" >> inex.html
```

//浏览器测试

Ningx集群环境搭建

反向代理(nginx \longleftrightarrow apache)

位于Internet的客户端通过企业边界设备获取企业内部的web服务器资源.

Ningx集群环境搭建

反向代理(**nginx**←→**apache**)

实验所需环境

一台带有**Nginx**虚拟机做反向代理服务

一台带有**apache**虚拟机做后台**www**服务

Ningx集群环境搭建

反向代理(nginx<—>apache)

示意图:



Ningx集群环境搭建

反向代理(nginx<—>apache)

编辑nginx反向代理服务器的配置文件

```
#vim /etc/nginx/nginx.conf
```

在server区段下增加内容:

```
server{  
  
    listen 80 default_server;  
  
    listen [::]:80 default_server;  
  
    server_name test,niliu.edu;
```

Ningx集群环境搭建

反向代理(nginx<—>apache)

```
proxy_redirect      off;

proxy_set_header    X-real-IP
$remote_addr;

proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;

proxy_set_header    Host$http_host;

location / {

    proxy_pass http://test1.niliu.edu/;

}

}
```

Ningx集群环境搭建

反向代理(nginx<—>apache)

#systemctl restart nginx

修改apache服务

#vim /etc/httpd/conf/httpd.conf

修改第196行:

**LogFormat "%{X-Forwarded-For}i" %l %u %t \"%r\"
%>s %b \"%{Referer}i\" \"%{User-Agent}i\" combined**

Ningx集群环境搭建

反向代理(nginx<—>apache)

#systemctl restart httpd

浏览器测试

http://test.niliu.edu

应该显示test1主机的内容

Ningx集群环境搭建

反向代理及负载均衡(**nginx**←→**apache***3)

所需环境:

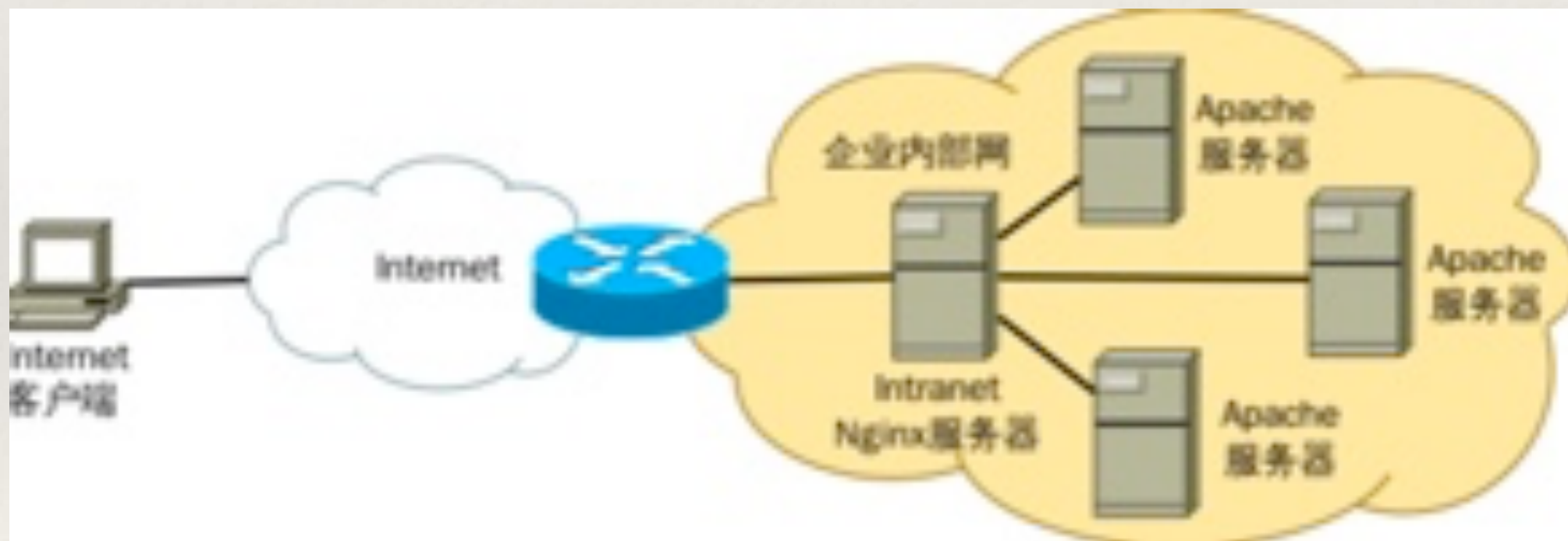
nginx反向代理服务器1台

三台**apache**服务并运行,每台页面内容不一样以便区分

Ningx集群环境搭建

反向代理及负载均衡(nginx<—>apache*3)

❖ 示意图



Ningx集群环境搭建

反向代理及负载均衡(nginx<—>apache*3)

配置Nginx

#vim /etc/nginx/nginx.conf

在http区段,设定负载均衡组

Ningx集群环境搭建

反向代理及负载均衡(nginx<—>apache*3)

```
httpd{
```

```
    upstream backends{
```

```
        server test1.niliu.edu:80 weight=3 max_fails=3 fail_timeout=20s;
```

```
        server test2.niliu.edu:80 weight=2 max_fails=3 fail_timeout=20s;
```

```
        server test3.niliu.edu:80 weight=1 max_fails=3 fail_timeout=20s;
```

```
    }
```

```
}
```

Ningx集群环境搭建

反向代理及负载均衡(nginx<—>apache*3)

设置反向代理至负载均衡组

```
server{  
    listen 80 default_server;  
    listen [::]:80 default_server  
    server_name test.niliu.edu
```

Ningx集群环境搭建

反向代理及负载均衡(nginx<—>apache*3)

```
proxy_redirect      off;

proxy_set_header    X-real-IP
$remote_addr;

proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;

proxy_set_header    Host$http_host;

location / {

    proxy_pass http://backends;

}

}
```

Ningx集群环境搭建

反向代理及负载均衡(**nginx \longleftrightarrow apache*3**)

测试

#systemctl restart nginx

浏览器测试

http://test.niliu.edu 多次执行后会执行出各个**apache**的内容

Ningx集群环境搭建

反向代理及负载均衡(nginx<—>apache*3)

配置文件中添加如下字段即可实现nginx负载故障迁移

location / {

proxy_pass http://backends;

**proxy_next_upstream http_500 http_502 http_503 error timeout
invalid_header;** //表示当nginx发现服务节点错误后会自动
将请求转发到负载均衡集群组中的另一个服务器上,实现故障转移

include /etc/nginx/proxy.conf //设定反向代理配置,优化代理能力

}

Ningx集群环境搭建

Nginx+keepalived搭建高可用集群

使用场景:

nginx做负载均衡，来达到分发请求的目的，但是不能很好的避免单点故障，
nginx务器挂点了，那么所有的服务也会跟着瘫痪
keepalived+nginx，就可以很好的解决这一问题

Ningx集群环境搭建

反向代理及负载均衡(nginx<—>apache*3)

在proxy.conf中可有以下内容:

proxy_redirect off; //是否对发送给客户端的URL进行修改

例如:

location: http://test.abc.com:9080/abc.html 。因为 **nginx** 服务器侦听的是 **80** 端口, 所以这样的 **URL** 给了客户端, 必然会出错. 针对这种情况, 加一条 **proxy_redirect** 指令: **proxy_redirect http://test.abc.com:9080/ /**, 把所有“**http://test.abc.com:9080/**”的内容替换成“**/**”再发给客户

Ningx集群环境搭建

反向代理及负载均衡(nginx<—>apache*3)

设置由后端服务器获取用户的主机名或真实 IP 及代理的 IP

```
proxy_set_header Host $host;
```

```
proxy_set_header X-Real-IP $remote_addr
```

```
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
```

```
client_body_buffer_size 128k;    //用于指定客户端请求主体缓冲区大小，可理解 为先保存到本地在传送给用户
```

```
proxy_connect_timeoute 90        // 表示与后端服务器连接超时时间，  
从发起握手开 始，单位为秒
```

Ningx集群环境搭建

反向代理及负载均衡(nginx<—>apache*3)

proxy_send_timeout 90; //后端服务器传回数据的超时时间 ,如超过则断开 连接

proxy_read_timeoute 90; //nginx 等待后端服务器处理的等候超时时间

proxy_buffer_size 4k; /* 设置缓冲区大小 (等于 proxy_buffers 设置的大 小)

proxy_buffers 4 32k; // 设置缓冲区的数量及大小

proxy_buffersx2 proxy_busy_buffers_size 64k; //设置系统繁忙时可以使用的
proxy_buffers 大 小

proxy_temp_file_write_size 64k; // 指定 proxy 缓存临时文件的大小

Ningx集群环境搭建

Nginx+keepalived搭建高可用集群

原理:

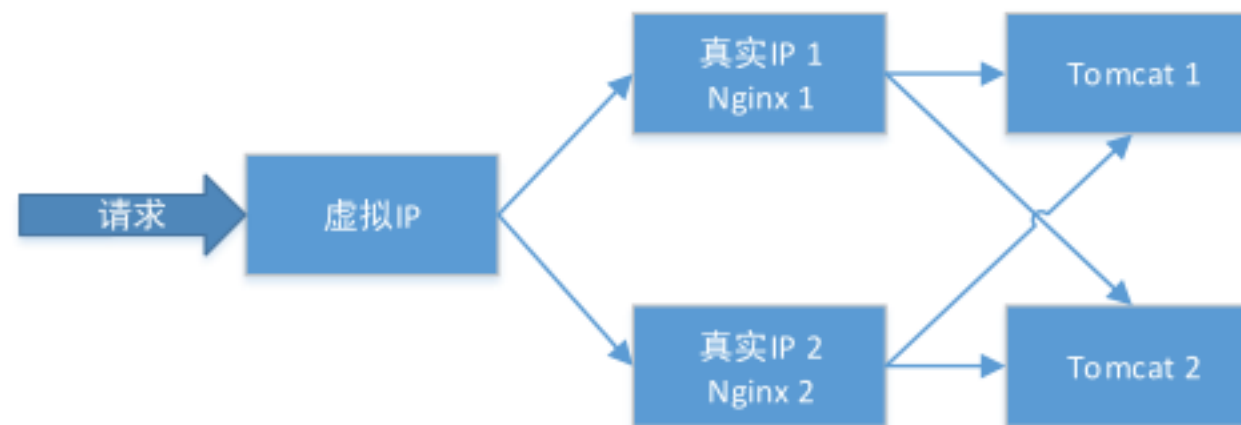
Keepalived 是一种高性能的服务器高可用或热备解决方案，Keepalived 可以用来防止服务器单点故障的发生，通过配合 Nginx 可以实现 web 前端服务的高可用

利用Keepalived 的 VRRP(Virtual Router Redundancy Protocol) 协议，来实现高可用性(HA)。

Ningx集群环境搭建

Nginx+keepalived搭建高可用集群

示意图



使用Keepalived来实现Nginx高可用

Ningx集群环境搭建

Nginx+keepalived搭建高可用集群

准备环境

两台虚拟机如:

192.168.0.121和192.168.0.122

安装nginx所需要的运行库

```
yum install gcc gcc -c++ zlib zlib-devel pcre pcre-devel openssl  
openssl-devel
```

Ningx集群环境搭建

Nginx+keepalived搭建高可用集群

gcc :gcc编译器

gcc -c++ :c++编译器用于编译nginx中的http模块

pcre, pcre-devel: pcre库,nginx中rewrite模块需要

openssl,openssl-devel:用于支持在ssl协议上传输HTTP

Ningx集群环境搭建

Nginx+keepalived搭建高可用集群

安装

#wget <http://nginx.org/download/nginx-1.9.5.tar.gz>

#获取nginx源码包

#tar -zxvf nginx-1.9.5.tar.gz //解压

#cd /nginx-1.9.5 //进入源码目录

#./configure 编译

#make && make install //编译并安装

#systemctl start nginx.service //启动服务

Ningx集群环境搭建

Nginx+keepalived搭建高可用集群

安装keepalived:

wget <http://www.keepalived.org/software/keepalived-1.2.22.tar.gz> //下载到本地

#mkdir -p /alidata/server/keepalived //创建目录

#tar -zxvf keepalived-1.2.22.tar.gz //解压缩

#cd keepalived-1.2.22

#./configure //编译

#make && make install //安装

#systemctl start keepalived.service //启动服务

Ningx集群环境搭建

Nginx+keepalived搭建高可用集群

修改配置文件:

```
#vim /etc/keepalived/keepalived.conf
```

从第8行开始

```
vrrp_script chk_nginx {
```

```
script "/alidata/server/keepalived/nginx_check.sh" #运行脚本，脚本内容下面有，就是起到一个nginx宕机以后，  
自动开启服务
```

```
interval 2      //检测时间间隔
```

```
weight -20      //如果条件成立的话，则权重 -20
```

```
}
```

Ningx集群环境搭建

Nginx+keepalived搭建高可用集群

```
vrp_instance VI_1 {  
    state MASTER           //来决定主从,从机器为BACKUP  
  
    interface eno1677736    //绑定虚拟 IP 的网络接口，根据自己的机器填写  
  
    virtual_router_id 121   //虚拟路由的 ID 号， 两个节点设置必须一样  
  
    mcast_src_ip 192.168.43.121 #填写为本机ip  
  
    priority 100            //节点优先级,主要比从节点优先级高  
  
    nopreempt              //优先级高的设置 nopreempt 解决异常恢复后再次抢占的问题  
  
    advert_int 1           // 组播信息发送间隔，两个节点设置必须一样，默认 1s  
  
    authentication {  
        auth_type PASS  
        auth_pass 1111  
    }  
}
```

Ningx集群环境搭建

Nginx+keepalived搭建高可用集群

```
track_script {
```

```
chk_nginx          //执行 Nginx 监控的服务
```

```
}
```

```
virtual_ipaddress {
```

```
192.168.43.10      // 虚拟ip,也就是解决写死程序的ip怎么能切换的ip,也可扩展，用途广泛。可配置多个。
```

```
}
```

```
}
```

从机与主机配置文件相同,网络地址以及权重必须更改

Ningx集群环境搭建

Nginx+keepalived搭建高可用集群

监控脚本内容

```
#!/bin/bash
```

```
A=`ps -C nginx --no-header lwc -l`  
if [ $A -eq 0 ];then
```

```
/alidata/server/nginx/sbin/nginx
```

```
sleep 2
```

```
if [ `ps -C nginx --no-header lwc -l` -eq 0 ];then
```

```
killall keepalived
```

```
fi
```

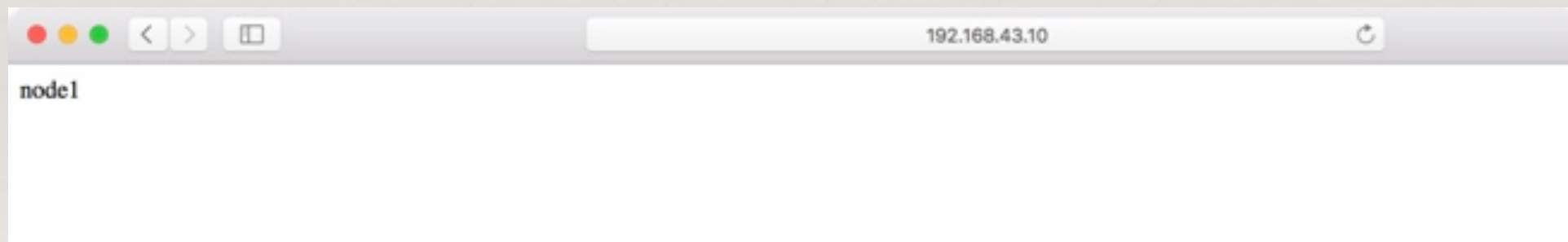
```
fi
```

保存并重启服务

Ningx集群环境搭建

Nginx+keepalived搭建高可用集群

打开浏览器输入虚拟IP地址:



Ningx集群环境搭建

Nginx+keepalived搭建高可用集群

此时，我停掉其中一台的**keepalived**服务



Ningx集群环境搭建

Nginx+keepalived搭建高可用集群

此时，我们已完成标题内容。由此可以扩展出更多的高可用架构,并不局限于课件内容.