operation 04

=============================================

**一，nginx问题处理**

1. **缓存文件在客户端**

**用户如果反复访问服务器的相同文件可以缓存到客户机，避免**

**服务器反复传送相同文件，节约时间**

打开配置文件，在默认的location下面添加一个location

location ~\* \.(jpg|html|txt|mp3)$ { //当发现用户访问的是

以.jpg或者.html等等结尾的页面时

expires 30d; //就把上述访问的文件在客户机缓存30天

}

sbin/nginx -s reload //配置完后重加载配置文件

使用火狐浏览器清空历史记录，然后访问192.168.2.5/test.jpg

然后地址栏输入about:cache

查看disk文件的列表，找到被访问文件看最后倒数第2列(Expires)

信息显示多久过期可以看到该文件的过期时间是一个月之后，说明

缓存成功。

----------------------------------------------------------

**2，支持超长地址栏**

**默认情况下nginx无法支持长地址栏，会报414错误**

打开配置文件,在默认的虚拟主机上方添加两行

client\_header\_buffer\_size 200k; //第一行表示,用户访问网站的头部信

息(包含地址栏)长度支持200k大小

large\_client\_header\_buffers 4 200k; //第二行表示,如果200k不够,再

给4个200k

sbin/nginx -s reload //重加载配置

然后到lnmp\_soft目录找到buffer.sh 脚本并运行(该脚本里的4.5可以修改为2.5),

看不到414报错即可./buffer.sh //执行测试脚本，可以支持超长地址栏并看到页

面内容，而不是414报错

----------------------------

上述脚本中地址的重新赋值过程如下，最终会产生常超地址

URL=http://192.168.4.5/index.html?

URL=http://192.168.4.5/index.html?v1=1

URL=http://192.168.4.5/index.html?v1=1v2=2

URL=http://192.168.4.5/index.html?v1=1v2=2v3=3

URL=http://192.168.4.5/index.html?v1=1v2=2v3=3.......................v5000=5000

-------------------------------------------------------------------------

**3，优化nginx并发**

用proxy 和web1

**防火墙与selinux都关闭**

[root@web1 ~]# yum -y install httpd-tools

ab -c 200 -n 200 <http://192.168.2.5/> //用压力测试工具模拟200人，每人1次，一共200次对2.5的网站发起访问，成功

ab -c 2000 -n 2000 <http://192.168.2.5/> //模拟2000人，每人1次，一共2000次

对2.5的网站发起访问，失败

**接下来进行优化**

在proxy主机：

打开nginx配置文件修改第3行，第13行-

worker\_processes 2; //开启的nginx进程数量，通常是随cpu的核心数一致

worker\_connections 50000; //每个nginx进程支持的并发访问量

sbin/nginx -s reload

proxy与web1都执行以下命令

ulimit -n //查询系统打开文件数量的大小

ulimit -n 100000 //临时定义文件可以同时被打开的次数为10万

回到web1主机执行命令ab -c 2000 -n 2000 http://192.168.2.5/ 看到100%则

成功

--------------------------------------

vim /etc/security/limits.conf //打开配置文件实现永久修改

修改第53、54行

\* soft nofile 100000

\* hard nofile 100000

-----------------------------------------------------------------------------------

**二，解决集群主机过多而导致用户重复登陆网站的问题**

**在一个集群中，如果网站需要用户输入用户名和密码登陆之后才能继续访问，那么当用户登陆其中一台集群主机之后随着继续访问页面，请求可能被代理服务器轮询到另外一台服务器上，那么对于另外一台服务器来说用户并没有登陆，想查看登陆之后的页面还需要再次登陆，这样集群主机越多需要客户重复登陆的次数就越多**

**1，按照下列需求部署好环境**

proxy web1 web2

nginx lnmp lnmp

在proxy主机家目录下将lnmp\_soft.tar.gz拷贝到web1

[root@proxy ~]# scp lnmp\_soft.tar.gz 192.168.2.100:

首先在web1主机：

yum -y install gcc make pcre-devel openssl-devel

[root@web1 ~]# tar -xf lnmp\_soft.tar.gz

[root@web1 ~]# cd lnmp\_soft/

tar -xf nginx-1.17.6.tar.gz

cd nginx-1.17.6/

./configure //配置

make //编译

make install //安装

yum -y install mariadb mariadb-server mariadb-devel //安装数据库

相关软件包

yum -y install php php-mysql php-fpm //安装php相关软件包

systemctl start mariadb

systemctl start php-fpm

[root@web1 nginx]# vim conf/nginx.conf //修改配置文件,实现动静分离,

修改65~71行为以下状态

vim conf/nginx.conf //修改配置文件,实现动静分离,修改65~71行为以下状态

location ~ \.php$ {

root html;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

# fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;

include fastcgi.conf;

}

配置好lnmp之后

[root@web1 nginx]# systemctl stop httpd

[root@web1 nginx]# sbin/nginx //开启服务

配置好lnmp之后

cd ~/lnmp\_soft/php\_scripts/

tar -xf php-memcached-demo.tar.gz //释放带登录功能的网页

cp -r php-memcached-demo/\* /usr/local/nginx/html/ //拷贝页面

到nginx中使用火狐浏览器访问http://192.168.2.100/index.php 可以

看到有登录界面的网页

**web1配置好之后，web2也按照一模一样的过程配置，防火墙都要关闭**

**2，然后在proxy主机配置集群，注意该主机不能有动静分离的配置**

upstream web { //在默认的虚拟主机上面添加集群

server 192.168.2.100:80;

server 192.168.2.200:80;

}

location / {

proxy\_pass http://web; //然后在默认的虚拟主机中的location里调用集群

sbin/nginx -s reload //重新加载配置文件

http://192.168.2.5/index.php //之后测试效果，不断刷新页面，会看到web1与

web2的登录界面(需要提前在web1与web2的index.php页面进行标记)

注意！proxy主机不要配置动静分离

---------------------------------------------------------------------

**Session 存储在服务器端，保存用户名、登录状态等信息**

**Cookies 由服务器下发给客户端，保存在客户端的一个文件**

**保存的主要内容是sessionID**

index.php 登录前页面

home.php 登录后页面

客户 Cookies ------ proxy ------ web1 Session

------ web2 Session

目前环境搭建好之后可以按以下步骤测试：

删除web1与web2的session文件

rm -rf /var/lib/php/session/\*

删除火狐浏览器的历史记录(主要是cookies文件)

http://192.168.2.5/index.php //登录2次才能成功，目前登录2次是正常现象！！

------------------------------------------------------------------

**上述实验由于web1与web2都是在各自的/var/lib/php/session目录中存储session，所以造成客户需要重复登录，为了统一session存储的位置（该存储方式通常被称为session共享），需要安装专门的数据库工具**

**memcache 可以利用内存读写数据的数据库服务**

**proxy主机：**

yum -y install memcached telnet //安装软件包

systemctl start memcached //开启服务

telnet 127.0.0.1 11211 //连接memcache，进行检测

之后测试memcached：

set abc 0 200 3 //创建变量abc(如果abc已经存在就是覆盖)，0是不压

缩数据，数据存储时间200秒，存3个字符，回车之后比如输入xyz就是存储

这3个字符

get abc //获取变量abc

replace abc 0 200 3 //覆盖abc，此时变量abc必须存在

delete abc //删除abc

add xyz 0 200 3 //添加abc变量，如果abc已经存在则会添加失败

flush\_all //删除所有数据

quit //退出

**到web1与web2主机修改存储session文件的位置**

vim /etc/php-fpm.d/www.conf //到最后一页

php\_value[session.save\_handler] = memcache //这里改成memcache，表示session

的存储不在是本地的普通文件，而是去找memcache

php\_value[session.save\_path] = tcp://192.168.2.5:11211 //这里的路径修改为

安装了memcache服务的服务器地址与端口

yum -y install php-pecl-memcache //安装php与memcached服务关联的软件包

systemctl restart php-fpm

**所有主机关闭防火墙和selinux**

**最后清空浏览器的历史记录，再访问http://192.168.2.5/index.php仅仅登录一次即**

**可成功**

------------------------------------------------------

**部署环境任务**

**找码云中 “环境准备” 文件，是一个压缩包**

**里面按照自己真机的环境(windows或者linux)然后找到 AUTOMATION的 day01-day03环境准备，按照文档中的描述一步一步部署下周课程的环境**